

Resum

ETSEIB Calendar és un projecte que té com a objectiu desenvolupar una aplicació que permeti la creació d'un sistema d'organització d'horaris de pràctiques de laboratori per els alumnes del GETI. Aquesta aplicació es basarà en el principi de la creació d'arxius compiladors d'esdeveniments que avui en dia són utilitzats en varies plataformes de correu, com Gmail o Outlook. A la totalitat de l'arxiu se li donarà format iCalendar, que tindrà tota la informació de les pràctiques seleccionades emmagatzemada per una posterior importació a un gestor de calendaris. A partir d'una selecció d'assignatures i de grups, l'estudiant rebrà a canvi un arxiu que al importar-lo li permetrà aconseguir un calendari complet de totes les pràctiques de laboratori. A més a més, la plataforma permetrà personalitzar les notificacions de pràctica al seu gust, personalitzant amb quina antelació es realitzaran els avisos de pràctica. Utilitzant aquesta eina l'estudiantat de l'ETSEIB no oblidarà més les seves pràctiques de laboratori un cop estigui matriculat i podrà emprar aquest temps que abans passava apuntant-les en altres afers més productius.

Gràcies a l'aplicació Kivy com a mòdul de Python és desenvoluparà l'aplicació des de 0, partint dels mateixos principis amb els que treballa el portal d'assignatures i horaris de ETSEIB. A través d'una crida a la base de dades de les pràctiques de la ETSEIB, l'aplicació obtindrà els horaris de pràctiques de les diferents assignatures (actualitzades de manera permanent a la base de dades). Per tant, aquesta eina no té com a data de caducitat un quadrimestre, sinó que podria ser utilitzada permanentment, ja que no cal realitzar un manteniment. El fet de que l'aplicació sigui *responsive* a més a més, permetrà executar l'aplicació dins de pantalles de diferents mides (dins d'uns requisits mínims). Aquesta plataforma té com a objectiu final esdevenir pont per la evolució del portal d'assignatures i horaris a través els estudiants podrien simular els horaris (actualment ja ho fa) i a més a més podrien descarregar els arxius de calendari de les seves pràctiques al incorporar l'algorisme ETSEIB Calendar.

Sumari

RESUM	1
LLISTAT DE FIGURES	5
LLISTAT DE TAULES	6
LLISTAT D'EQUACIONS	7
1. INTRODUCCIÓ	8
1.1. Objectius del projecte	8
1.2. Abast del projecte	9
2. ARQUITECTURA D'ETSEIB CALENDAR	11
2.1. Com s'obté la informació?	12
2.2. Com es dona format a l'arxiu iCalendar?	13
2.3. Com s'aconsegueix integrar-ho en una app?	15
2.4. Esbós de la plataforma	15
3. CONCEPTES PREVIS	18
3.1. Python	18
3.1.1. Per què Python?	18
3.2. Kivy	19
3.3. Bases de dades	20
3.4. iCalendar	21
4. IMPLEMENTACIÓ D'ETSEIB CALENDAR	22
4.1. Arxicsmulti.py	22
4.1.1. Funcions arxicsmulti.py	24
4.2. Filtro.py	24
4.2.1. Funcions de Filtro.py	25
4.3. Main.py	25
4.3.1. Classe Welcome	27
4.3.2. Classe MainScreen	27
4.3.3. Classe SecondScreen	28
4.3.4. Classe AnotherScreen	28
4.3.5. Classe Root	29
4.3.6. Classe SimpleKivy	29
4.4. Taula resum funcions i arxius	30
5. KIVY LANGUAGE	32
6. CREACIÓ DE LA PANTALLA DE SELECCIÓ DE GRUPS	45
7. PLANIFICACIÓ ETSEIB CALENDAR	49

7.1. Planificació temporal.....	49
7.2. Planificació econòmica	51
8. PAQUETITZACIÓ DE L'APLICACIÓ	53
9. ETSEIB CALENDAR	55
9.1. Captures aplicació	55
9.2. Arbre d'errors	61
9.3. Tramitació de les sol·licituds de calendari	66
9.4. Exemple ETSEIB Calendar	67
9.5. Control estadístic	71
10. PROPOSTA DE CONTINUACIÓ DEL TREBALL	73
CONCLUSIONS	74
AGRAÏMENTS	75
BIBLIOGRAFIA	76
Referències bibliogràfiques.....	76
Importància de les referències	77

Llistat de figures

<i>Figura 2.2: Primer esquema funcionament ETSEIB Calendar.....</i>	<i>11</i>
<i>Figura 2.3: Esquema final funcionament ETSEIB Calendar</i>	<i>12</i>
<i>Figura 2.4: Esbós de la pantalla inicial.....</i>	<i>15</i>
<i>Figura 2.5: Esbós de la selecció d'assignatures.....</i>	<i>16</i>
<i>Figura 2.6: Esbós de selecció de grups.....</i>	<i>16</i>
<i>Figura 2.7: Esbós operació realitzada.....</i>	<i>17</i>
<i>Figura 3.1: Logotip de Python</i>	<i>18</i>
<i>Figura 3.2: Logotip de Kivy.....</i>	<i>19</i>
<i>Figura 7.1: Diagrama Gantt d'ETSEIB Calendar</i>	<i>49</i>
<i>Figura 9.1: Pantalla de càrrega d'ETSEIB Calendar</i>	<i>55</i>
<i>Figura 9.2: Primera versió pantalla inicial d'ETSEIB Calendar</i>	<i>56</i>
<i>Figura 9.3: Segona versió de la pantalla inicial d'ETSEIB Calendar.....</i>	<i>56</i>
<i>Figura 9.4: Tercera versió pantalla inicial d'ETSEIB Calendar.....</i>	<i>57</i>
<i>Figura 9.5: Versió final de pantalla inicial.....</i>	<i>57</i>
<i>Figura 9.6: Captura del Popup informatiu d'ETSEIB Calendar</i>	<i>58</i>
<i>Figura 9.7: Captura de de la primera pantalla de selecció d'assignatures d'ETSEIB Calendar.....</i>	<i>59</i>
<i>Figura 9.8: Captura de de la segona pantalla de selecció d'assignatures d'ETSEIB Calendar</i>	<i>59</i>
<i>Figura 9.9: Captura de de la pantalla de selecció de grups d'ETSEIB Calendar</i>	<i>60</i>
<i>Figura 9.10: Captura del Popup d'operació realitzada d'ETSEIB Calendar.....</i>	<i>61</i>
<i>Figura 9.11: Esquema de funcionament de la primera pantalla.....</i>	<i>62</i>
<i>Figura 9.12: Popup corresponent a l'error 1</i>	<i>62</i>
<i>Figura 9.13: Popup corresponent a l'error 2</i>	<i>63</i>
<i>Figura 9.14: Esquema de funcionament de les pantalles de selecció d'assignatures.....</i>	<i>63</i>
<i>Figura 9.15: Popup corresponent a l'error 3</i>	<i>64</i>
<i>Figura 9.16: Popup corresponent a l'error 4</i>	<i>64</i>
<i>Figura 9.17: Esquema de funcionament de la pantalla de selecció de grups.....</i>	<i>65</i>
<i>Figura 9.18: Popup corresponent a l'error 5</i>	<i>65</i>
<i>Figura 9.19: Popup corresponent a l'error 6</i>	<i>66</i>
<i>Figura 9.20: Primera pantalla amb dades inserides</i>	<i>67</i>
<i>Figura 9.21: Primer bloc d'assignatures amb seleccions</i>	<i>68</i>
<i>Figura 9.22: Segon bloc d'assignatures amb seleccions</i>	<i>68</i>
<i>Figura 9.23: Pantalla de selecció de grups amb grups inserits</i>	<i>69</i>
<i>Figura 9.24: Missatge d'operació realitzada amb èxit</i>	<i>69</i>
<i>Figura 9.25: Correu electrònic que es rebrà amb l'arxiu contenidor.....</i>	<i>70</i>
<i>Figura 9.26: Possibilitats que dona Outlook al rebre el correu electrònic</i>	<i>70</i>
<i>Figura 9.27: Avís que es rebrà amb l'antelació seleccionada</i>	<i>70</i>
<i>Figura 9.28: Exemple d'anàlisi estadístic que aporta Gmail Meter</i>	<i>71</i>
<i>Figura 9.29: Exemples de resums gràfics de Gmail Meter.....</i>	<i>72</i>

Llistat de taules

<i>Taula 4.1: Resum de les funcions de l'arxiu arxicsmulti.py.....</i>	<i>30</i>
<i>Taula 4.2: Resum de les funcions de l'arxiu filtro.py</i>	<i>30</i>
<i>Taula 4.3: Resum de les classes i funcions de l'arxiu main.py</i>	<i>31</i>
<i>Taula 7.1: Resum de costos associats al projecte</i>	<i>52</i>

Llistat d'equacions

<i>Equació 7.1: Cost del hardware</i>	<i>52</i>
---	-----------

1. Introducció

El projecte ETSEIB Calendar apareix amb l'objectiu d'arribar a ser una eina d'ús habitual per els alumnes de la escola, que ara mateix han d'organitzar el tema dels horaris de pràctiques d'assignatures d'una forma molt poc eficient. El fet de que aquest procés (habitual en moltes assignatures) esdevingui tan manual, provoca que els alumnes molts cops no organitzen les seves agendes i obliden sessions de laboratori on és obligatòria la seva presència. La creació d'aquesta eina serà a través del llenguatge Python, ja utilitzat en algunes assignatures durant el transcurs del GETI, i té com objectiu final, acabar essent un prototip de cara a un futur proper, en qual es podrà incorporar a la pàgina de web de ETSEIB. Després d'estudiar les diverses possibilitats per desenvolupar la plataforma es va decidir fer-ho a través del mòdul Kivy App de Python (Kivy s'ocuparà de la interfície gràfica) i diferents mòduls que permetin crear els arxius compiladors d'esdeveniments.

1.1. Objectius del projecte

El projecte té com a finalitat esdevenir una eina útil per la gestió de calendaris de pràctiques de laboratori, creant una interfície que permeti a l'alumne introduir les assignatures i grups desitjats, i retorni com a resultat un fitxer on estigui emmagatzemada tota la informació d'horaris dels grups seleccionats. Per tal de realitzar la simulació amb horaris de pràctiques actualitzats, s'accedirà a la base de dades d'on s'extreu tota la informació de la pàgina web de ETSEIB, permetent així la experiència de tractament de dades actualitzades i fiables. Els objectius (tant principals com secundaris), es poden veure sintetitzats a continuació:

Principals:

- Facilitar organitzar les pràctiques (GETI) de cara als estudiants de l'ETSEIB.
- Disseny d'una aplicació a través de Python, a través d'una Kivy App.
- Obtenir els horaris actualitzats de la base de dades amb la que treballa actualment la web de la escola.
- Crear un prototip de cara a una futura implementació a la web d'ETSEIB

Secundaris:

- Treballar a través d'una app web, tant en un PC com en un dispositiu Android o IOS.
- Automatitzar el processament de l'arxiu obtingut com a *output* de l'eina, ja que com es veurà en primera instància esdevindrà manual.



- Gestionar els avisos que es desitgen per les pràctiques (1 dia abans, 12 hores abans...).
- Personalitzar la interfície en una gama de colors positius o negatius, per tal de facilitar la seva utilització en tots el àmbits d'il·luminació.

1.2. Abast del projecte

Degut a que no es disposa de molt temps per la realització del projecte i que es treballarà amb una nova arquitectura de programació (mòdul Kivy) , cal dir que caldrà limitar el seu abast dintre d'uns marges específics. En primera instància, val a dir que ETSEIB Calendar s'ocuparà de la gestió de calendaris un cop l'alumne sàpiga a quin grup anirà, és a dir, no decidirà quin calendari és millor o pitjor ni tampoc indicarà solapaments entre assignatures (com fa actualment el simulador de calendaris de la web de l'escola). El fet de que això no ho integri el projecte és perquè ja existeixen altres eines que fan la mateixa funció i per tant, com l'objectiu final és complementar les eines existents actualment, no tindria sentit repetir una mateixa funció que realitza un altre mòdul de la web de la escola.

Com a segona limitació en l'abast del projecte, cal tenir en compte que ETSEIB Calendar gestionarà formularis, és a dir, no serà un procés automàtic on l'alumne posarà el seu usuari de e-secretaria i automàticament retornarà l'arxiu amb tots els horaris integrats. El que si farà l'eina és proporcionar formularis on l'usuari podrà seleccionarà les assignatures amb pràctiques de les que vulgui estar informat a través de la selecció de determinats grups. La utilització del mòdul Kivy potser limita el projecte també, ja que possiblement algunes de les funcions que es vulguin integrar dins de l'eina siguin impossibles d'aconseguir de forma trivial o la importació a altres plataformes pot arribar a ser molt complicada.

2. Arquitectura d'ETSEIB Calendar

Un cop tractats alguns conceptes previs, es pot iniciar l'explicació del desenvolupament de la plataforma ETSEIB Calendar. Tal i com s'ha parlat durant la introducció del projecte, aquest s'ocuparia de realitzar la gestió de formularis i la creació del document amb totes les dades necessàries per estar advertit de totes les pràctiques a les quals l'alumne estigui matriculat. Una explicació gràfica del procés seria la següent:

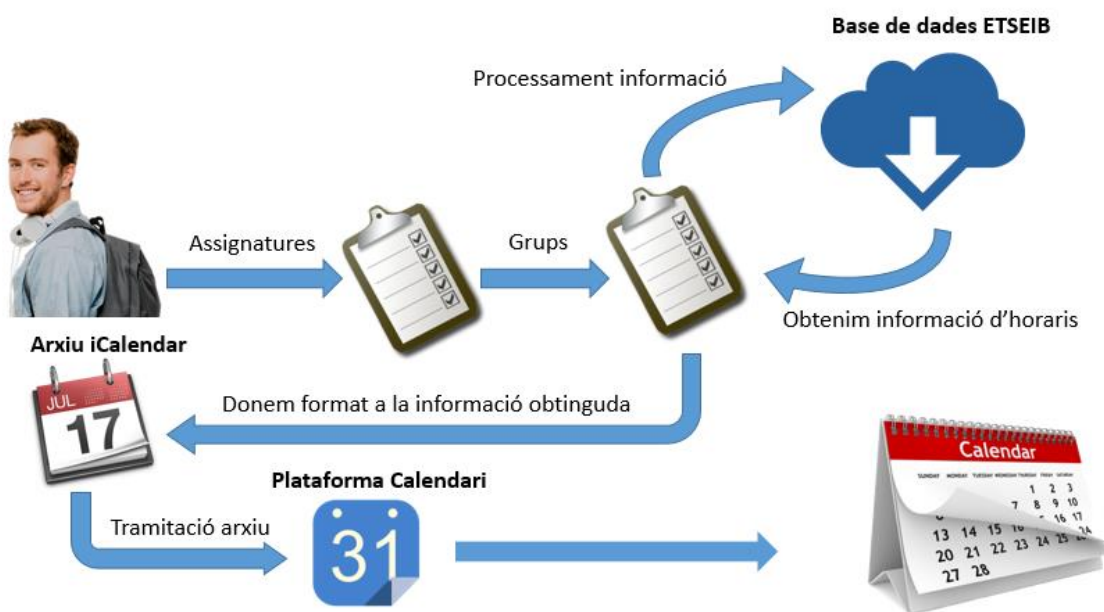


Figura 2.1: Primer esquema funcionament ETSEIB Calendar

L'objectiu d'aquest projecte és incorporar-ho a la pàgina web de la escola, concretament dins de l'eina del simulador de calendaris o del teu calendari personal. Però en una primera versió, l'estudiant seleccionaria les assignatures a través d'un formulari senzill, ja que no voldrà incorporar tots els horaris, sinó que només el de aquelles assignatures que realitzi durant aquell quadrimestre. Un cop inserides les assignatures, en una segona pantalla, l'alumne seleccionaria els grups als quals està matriculat, i un cop completades totes aquestes dades començaria la gestió dins de la base de dades de la ETSEIB. A través d'una consulta al document en format JSON amb totes les dades actualitzades de les pràctiques de laboratori, l'eina emmagatzemaria només les dades necessàries per l'estudiant (assignatures i grups seleccionats), compilant-les un mateix arxiu iCalendar que esdevindria el calendari total de pràctiques per l'alumne. Un cop realitzada la creació del document, en primera instància es va considerar que l'alumne integraria manualment l'arxiu dins de la seva plataforma, però estudiant un parell de simulacres que es van realitzar, el sistema no era del tot òptim, ja que l'alumne hauria de trobar el directori i accedir al document manualment dins la seva

plataforma. Així doncs, es va estudiar la possibilitat d'automatitzar el procés i millorar aspectes com ara la gestió dels avisos (inicialment estava pre-definida a 1 hora abans de la pràctica). Com a resultat es va obtenir aquest disseny:

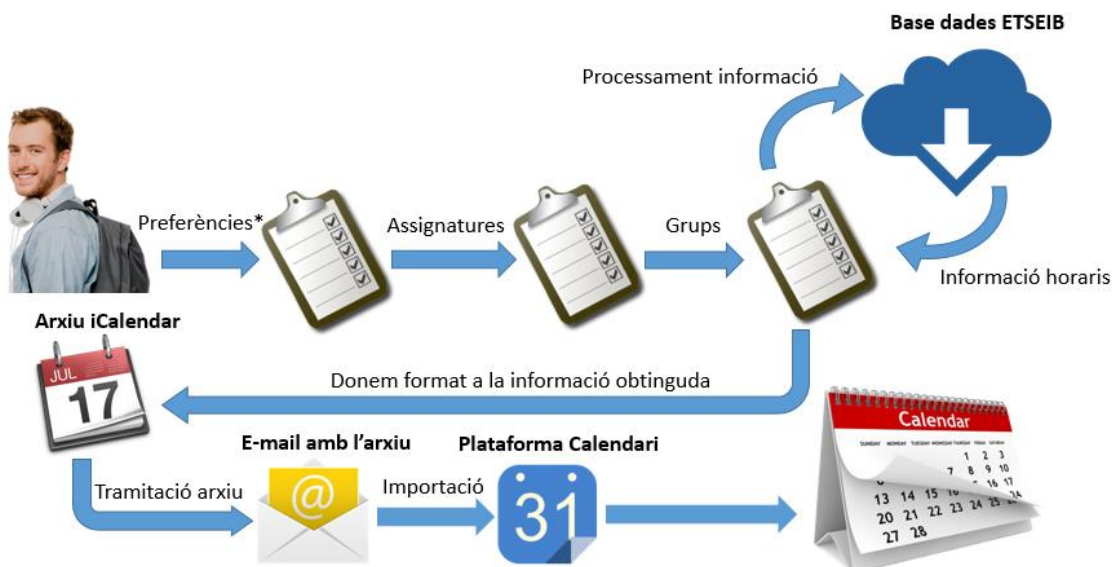


Figura 2.2: Esquema final funcionament ETSEIB Calendar

Com es pot observar, es va integrar un primer formulari on s'inseririen les preferències i una tramitació a través d'una plataforma e-mail. Les preferències són bàsicament amb quina l'antelació es vol rebre l'avís de la pràctica i el correu electrònic on es vol rebre l'arxiu ics. Durant l'explicació de cada mòdul i cada pantalla de l'aplicació es detallarà com s'han aconseguit aquestes funcionalitats.

2.1. Com s'obté la informació?

L'obtenció dels horaris de les pràctiques de laboratori es fa actualment a través de la web de l'escola. Integrat dins del portal de assignatures i horaris trobem que si seleccionem una sèrie d'assignatures i una sèrie de grups, podem obtenir una simulació total del calendari que tindriem amb els horaris seleccionats. Aquesta informació que es veu representada, és la mateixa informació amb la que treballa ETSEIB Calendar, però amb la diferència que en comptes de plasmar-la de forma gràfica, l'aplicació ETSEIB Calendar la integrarà en un arxiu iCalendar per tal de que l'usuari pugui aconseguir integrar el seu horari de pràctiques a la seva plataforma de calendaris. A través d'una consulta amb els serveis informàtics de la ETSEIB, es va proporcionar una adreça totalment actualitzada amb les pràctiques de laboratori en format json. Per tal d'aconseguir el calendari total, caldria navegar per els diferents nivells i emmagatzemar la informació de cada assignatura seleccionada per l'usuari, donant com a

resultat el calendari complert. Tal i com s'ha descrit abans, la base de dades esta composta per diferents nivells de calendaris i llistes, en las quals es troben per exemple el nom de l'assignatura, el codi o el quadrimestre en el qual es cursa. Un cop seleccionada l'assignatura, un nivell per sota es troben els diferents grups d'aquesta que en forma de diccionari emmagatzemaran la informació que ens interessa: dies de les diferents pràctiques, aula on es realitzen, hora d'inici i hora de finalització. L'explicació dels diferents nivells de l'arxiu json es pot veure a l'annex A.

2.2. Com es dona format a l'arxiu iCalendar?

Un cop s'obtenen i s'emmagatzemen les diferents dades per cada selecció de l'usuari (cada assignatura i grup) s'iniciarà el procés de donar format a l'arxiu iCalendar. A través del mòdul iCalendar(que es veurà en aparats posteriors de la memòria), es donarà format a un arxiu que inclourà la totalitat de les assignatures seleccionades amb els seus respectius grups amb el següent format:

- *BEGIN:VCALENDAR*. Indicarà que es crea un calendari, dintre d'un mateix arxiu .ics apareixerà només un cop.
- *BEGIN:VEVENT*. Indicarà la inicialització d'un esdeveniment (cada pràctica de cada assignatura és un esdeveniment).
- *SUMMARY*. Resum de l'esdeveniment.
- *DTSTART;VALUE=DATE-TIME*. Hora d'inici de l'esdeveniment. Caldrà indicar-ho en format UTC.
- *DTEND;VALUE=DATE-TIME*. Final de l'esdeveniment.
- *LOCATION*. Localització de l'esdeveniment. En el nostre cas serà l'aula que correspon a la pràctica de laboratori.
- *BEGIN:VALARM*. Inicialització de l'alarma de l'esdeveniment. Per cada esdeveniment caldrà inicialitzar l'alarma.
- *ACTION:EMAIL*. Acció que executarà l'alarma. En aquest cas e-mail significa que s'enviarà un e-mail al nostre correu indicant-nos l'existència de l'esdeveniment.
- *DESCRIPTION*. Descripció que apareixerà al e-mail.
- *TRIGGER*. Amb quina antelació s'executarà l'alarma respecte l'esdeveniment.

- *END:VALARM*. Finalització de l'alarma.
- *END:VEVENT*. Finalització de l'esdeveniment. Caldrà fer-ho per tancar l'esdeveniment e iniciar el següent.
- *END:VCALENDAR*. Tancament del calendari. Serà necessari per tancar l'arxiu iCalendar.

Per tal de veure un exemple il·lustratiu d'un arxiu iCalendar amb una pràctica de laboratori es pot consultar l'annex B. A continuació es mostra un exemple més simple per tal de entendre el concepte dels arxius iCalendar:

BEGIN:VCALENDAR

VERSION:2.0

BEGIN:VEVENT

SUMMARY: Abraham Lincoln

UID:2008-04-28-04-15-56-62-@americanhistorycalendar.com

DTSTART:20080212

DTEND:20080213

DTSTAMP:20150421T141403

CATEGORIES:U.S. Presidents,Civil War People

LOCATION:Hodgenville\, Kentucky

DESCRIPTION:Born February 12\, 1809\nSixteenth President (1861-1865)

END:VEVENT

END:VCALENDAR

Aquest arxiu emmagatzema un esdeveniment amb data el 12 de Febrer de 2008 fins el dia 13 i amb títol Abraham Lincoln. Ha sigut extret directament de la pàgina oficial de *iCalendar*.

2.3. Com s'aconsegueix integrar-ho en una app?

El principi teòric de ETSEIB Calendar és senzill, aportar preferències com són el correu electrònic i quan que t'avisí la plataforma que tens una pràctica, seleccionar les assignatures de les que vols rebre notificacions de les pràctiques i finalment seleccionar els grups on cursares aquestes assignatures. A canvi l'usuari rebrà un arxiu al seu correu on s'indicarà els diferents passos per integrar-lo a la seva plataforma calendari. I tot això com s'integra en un mateix arxiu? A través de la creació d'un arxiu APK, format que tenen les aplicacions Android. Existeixen moltes plataformes que permeten passar un arxiu creat a través de la plataforma Python a un arxiu .apk (instal·lador en dispositius Android) però es va decidir utilitzar Buildozer, un mòdul de Python que permet crear a partir d'un arxiu main.py i els diferents mòduls que utilitza aquest document, una apk completa. Degut a que el desenvolupament de l'aplicació en la seva totalitat es realitzarà a través de Kivy i alguns altres mòduls de Python, Buildozer és totalment vàlid per la realització d'aquesta tasca. A més a més s'intentarà donar suport a aquesta aplicació dins de Windows, Mac i IOS, dotant de més varietat de dispositius al programari.

2.4. Esbós de la plataforma

Un cop definida la idea de l'aplicació cal estipular les diferents pantalles de les que disposarà. A través d'un esbós preliminar es començarà a estudiar la idea que es portarà a la realitat a través de l'eina Kivy. Tal i com s'ha parlat anteriorment, es demanarien una sèrie de preferències a la primera pantalla de l'aplicació, la qual per concepte seria semblant a :

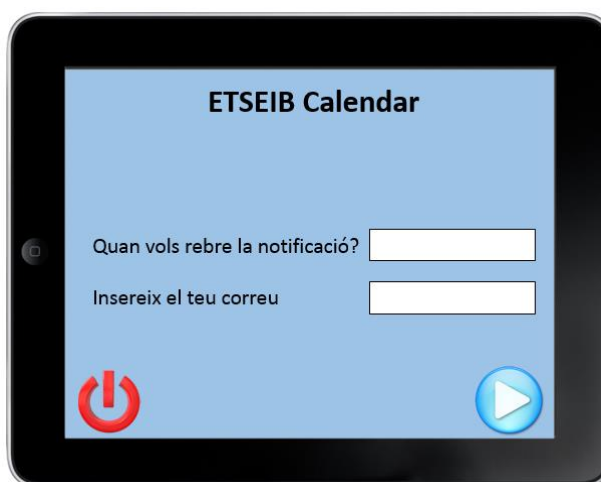


Figura 2.3: Esbós de la pantalla inicial

A través d'*inputs* com ara quan es vol rebre la notificació i l'e-mail on es vol rebre l'arxiu iCalendar, es prepararà l'obtenció del calendari. En cas de que l'usuari no completi els *inputs* l'aplicació no permetria avançar a la següent pantalla, la de selecció d'assignatures:

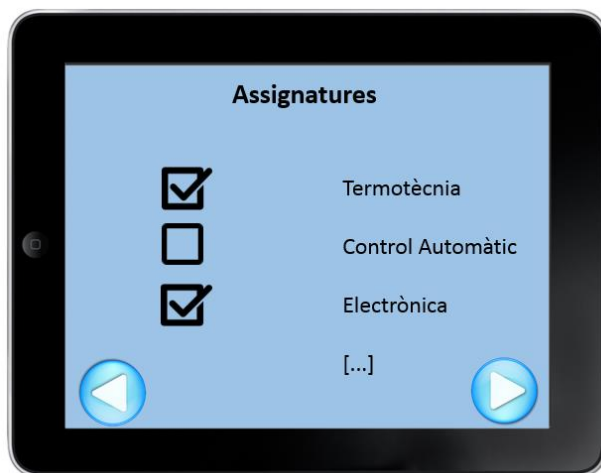


Figura 2.4: Esbós de la selecció d'assignatures

Donat que existeixen tretze assignatures que disposen de pràctiques de laboratori, es mostraran dos pantalles totes elles, set a la primera pantalla i 6 a la segona. L'usuari podrà seleccionar aquelles que actualment està cursant o de les que vol saber els calendaris i passar a la següent pantalla on seleccionarà els grups:



Figura 2.5: Esbós de selecció de grups

Cal dir que l'usuari de l'aplicació haurà d'introduir grups correctes dins dels *inputs* ja que en cas contrari no es permetrà inicialitzar el procés d'obtenció del calendari ical. Un cop introduïts al prémer el botó "Executa" si tot el procés s'ha pogut realitzar correctament, es notificarà d'aquesta manera a l'usuari:



Figura 2.6: Esbós operació realitzada

Cal dir però que caldrà protegir l'aplicació d'una sèrie d'errors com ara:

1. No completar els *inputs* correctament. El fet de inserir un correu no vàlid comportarà un error a la hora d'enviar l'arxiu calendari.
2. No seleccionar cap assignatura. L'objectiu de l'aplicació es donar calendaris d'assignatures, per tant caldrà seleccionar almenys una.
3. No inserir grups vàlids. L'usuari podria inserir grups invàlids o deixar camps sense emplenar. En el cas que passi això no s'executaria l'aplicació fins que s'integrin camps vàlids.
4. Necessitat de connexió amb la xarxa. L'aplicació treballa constantment amb dades o correus electrònics, per tant serà necessària connexió a Internet i no funcionarà si no disposa d'aquesta.

3. Conceptes previs

En aquest bloc del projecte, es presentaran alguns conceptes que serviran per entendre en la seva totalitat el funcionament de l'eina ETSEIB Calendar i el seu procés d'obtenció de les dades que es plasmaran. Fins ara s'han tractat temes merament teòrics, sense arribar a tocar *script* de Python. A través d'una petita introducció dels quatre principals conceptes profunds (programació Python, programació Kivy, gestió de bases de dades, iCalendar), l'explicació posterior de l'obtenció de l'eina esdevindrà més entenedora, així com la comprensió dels enllaços entre els diferents mòduls dels que estarà composta l'aplicació.

3.1. Python

Python és un llenguatge de programació dinàmic que permet el desenvolupament de "*scripts*". Aquests "*scripts*" al agrupar-se permetran desenvolupar programes tot lo complexes que el desenvolupador desitgi. A través de l'enllaç de diverses funcions, de la creació de condicions per tal de dur a terme una tasca (if, while, for..) o l'ús de variables de caires molt diferenciats (tuples, diccionaris, *strings*, llistes) , s'obtindrà el codi que permetrà realitzar les tasques precisades. Des de la seva aparició al 1989, el programa s'ha anat desenvolupant fins arribar als nostres dies, on grans empreses com per exemple Google ho integren com a plataforma de programació principal per les seves aplicacions AdWorks, Youtube...



Figura 3.1: Logotip de Python

3.1.1. Per què Python?

Els fets principals que ha comportat l'ús de Python pel desenvolupament d'ETSEIB calendar són els següents:

- Versatilitat d'ús. El desenvolupador decideix fins a quin nivell de complexitat vol arribar en el seu codi, adaptant-se així al nivell de l'usuari.
- Open-Source. El software de python és de caire obert, per tant no cal comprar llicències ni plataformes per tal d'utilitzar-ho.
- Ús en assignatures. Durant el transcurs del grau de tecnologies industrials, s'utilitza el

programari Python en algunes assignatures, com ara fonaments d'Informàtica o Informàtica. A més a més, es poden escollir algunes assignatures optatives on també es donarà ús a aquesta plataforma.

-Facilitat de creació d'aplicacions. A través de la incorporació de mòduls (pygame, math, json...), es faciliten moltes tasques de programació i la creació d'aplicacions, per exemple. Per crear l'aplicació ETSEIB Calendar s'ha decidit adoptar el mòdul Kivy, com es veurà en els següents punts del treball.

3.2. Kivy

Kivy és un programari que pot ser instal·lat en Linux, Windows, OS X, Android i IOS, i que permet la creació d'aplicacions de manera senzilla. Per tal de trobar un punt de comparació amb Python, amb el que també es podria crear una aplicació (mòdul pygame), cal dir que el llenguatge de Kivy és molt senzill i es poden crear a més a més, aplicacions a partir de la distribució d'uns *widgets* pre-definits, els quals permeten la integració d'elements a la teva aplicació de manera senzilla. Un exemple seria la creació d'un botó, en Python hauríem de crear un rectangle a la coordenada desitjada, fer que s'actualitzés l'estat de l'aplicació en quant es cliqués per exemple, i a més a més que el botó reaccions canviat de color quan l'usuari interactua amb ell. En canvi, amb el mòdul Kivy seria tan senzill com incorporar un *widget Button* a la coordenada on desitgessim afegir-ho, i indicar quina funció vols que fes a la hora de ser pitjat. A més a més, la interfície Kivy permet treballar en pantalles adaptatives (o *responsive*), les quals modifiquen la mida de tots els seus elements depenent del dispositiu on es visualitza l'aplicació.



Figura 3.2: Logotip de Kivy

Com es pot veure, Kivy té capacitat de sobres per desenvolupar l'aplicació ETSEIB Calendar. Potser el principal inconvenient és el no haver treballat mai amb aquest tipus de programari, cosa que comportarà avançar al principi una mica més lent del que es podria fer amb el mòdul *pygame* de Python.

3.3. Bases de dades

Com a tercer mòdul important de l'aplicació ETSEIB Calendar es troba l'obtenció d'informació a partir de bases de dades. Per tal de simular un aplicatiu totalment real, es va decidir contactar amb els serveis informàtics de la universitat amb l'objectiu de treballar de la mateixa manera que ells ho fan per simular calendaris o horaris a la pàgina web de l'escola. Després de comentar l'objectiu del projecte i els camps desitjats per la simulació, van crear una direcció url que contindria tota la informació en format Json. Aquesta informació posteriorment hauria de ser treballada, ja que s'emmagatzema tota la informació en forma de diccionaris i llistes en diferents nivells. Per tal d'entendre una mica més de la informació que gestiona l'aplicatiu ETSEIB Calendar, és necessari saber la distribució de l'arxiu JSON, que és la següent:

- Llista on la distribució és primer element GETI, segon element GEQ i tercer element GEM.
- Cada grau esta integrat en un diccionari, o es troben les variables *degree*, *degree name* i *subjects*.
- Si accedim a la *key subjects*, entrem en una altre llista integrada per diferents diccionaris, on cadascú conté la informació d'una assignatura amb pràctiques amb laboratori (13 en total).
- Dintre del diccionari de l'assignatura trobem dades com per exemple el codi d'assignatura, el nom en català, el quadrimestre al qual correspon i per últim els diferents grups de pràctiques integrats dins d'una llista.
- La distribució dels grups de pràctiques és similar a la de les assignatures, ja que és una llista amb diccionaris integrats dins. Dintre d'un diccionari trobarem totes les pràctiques d'un grup determinat, dins d'una altre llista de diccionaris (un per cada pràctica).
- Finalment a l'últim nivell de diccionaris es troba una sessió de pràctica, on podem accedir al dia, la hora de inici i de finalització i l'aula on es desenvoluparà la sessió.

A través de la navegació entre els diferents nivells d'aquest document com es pot veure podem obtenir tota la informació necessària per la creació d'un horari personalitzat. Per tal d'entendre el funcionament a través d'un cas pràctic, revisar annex A. Un exemple d'un document en format json seria el següent:

```
{
```

```

    "glossary": {
      "title": "example glossary",
      "GlossDiv": {
        "title": "S",
        "GlossList": {
          "GlossEntry": {
            "ID": "SGML",
            "SortAs": "SGML",
            "GlossTerm": "Standard Generalized
Markup Language",
            "Acronym": "SGML",
            "Abbrev": "ISO 8879:1986",
            "GlossDef": {
              "para": "A meta-markup language, used to create
markup languages such as DocBook.",
              "GlossSeeAlso": ["GML",
"XML"]
            },
            "GlossSee": "markup"
          }
        }
      }
    }
  }
}

```

Com es pot veure consta de diferents nivells, alguns d'ells són diccionaris i altres són llistes. Visualment no aporta molta informació, però a través de plataformes es pot aconseguir ordenar-ho d'una manera més entenedora. A través de la navegació entre nivells la plataforma ETSEIB Calendar podrà accedir a les assignatures seleccionades per l'usuari.

3.4. ICalendar

ICalendar és un format determinat de text (.ics) que permet emmagatzemar i enviar diferents esdeveniments i tasques a través d'internet, ja sigui via email o compartint els arxius amb aquesta determinada extensió. A través de la incorporació de diferents paràmetres, es pot aconseguir el mateix resultat que s'obtindria creant-ho manualment a través de plataformes web com Google Calendar o Outlook, d'entre d'altres. Per tal de compilar tota la informació de les diferents pràctiques es va decidir crear un arxiu iCalendar, ja que permetria la incorporació de tots els horaris dins d'un mateix arxiu. Gràcies a la seva compatibilitat amb Python, es pot treballar la creació d'arxius automàticament, creant diferents paràmetres com ara el nom de l'esdeveniment, hora i dia d'inici, avisos o ubicació. Un cop creat l'arxiu es pot importar a diverses plataformes de calendaris diferents, però aquest projecte únicament estudiarà les més usades actualment, Google Calendar i Outlook. Gràcies a la importació en aquestes plataformes, es podrà aconseguir una creació d'un calendari en el nostre smartphone, ja que Google Calendar i Outlook tenen versions adaptades a Android i IOS.

4. Implementació d'ETSEIB Calendar

En aquest punt del treball es començarà a estudiar com es va desenvolupar ETSEIB Calendar i tots els mòduls que el componen. S'intentaran tractar tots els diferents arxius de la mateixa manera: primer s'explicarà la part que gestionen de l'aplicació, segon s'estudiarà els mòduls de Python que cal importar en l'arxiu en qüestió i finalment es comentaran les classes i funcions que el componen. Primer s'estudiaran els mòduls que són una mica més intangibles, és a dir, aquells que no ens aporten una sortida visual, si més no, no veiem el resultat directament des de la pantalla. Un cop estudiats aquests mòduls es podrà entrar més fàcilment al desenvolupament teòric de la interfície gràfica de ETSEIB Calendar, que alterna *Kv language*¹ i Python.

4.1. Arxicsmulti.py

El mòdul `arxicsmulti.py` va ser el primer en esser desenvolupat i gestiona des de la creació del calendari `.ics` fins el seu enviament per correu electrònic. Inicialment el mòdul que s'encarregava d'enviar el correu electrònic estava separat al de creació del calendari, però es va decidir incorporar-los en un de sol per tal d'evitar realitzar passos entremetjats (com ara crear un arxiu i després importar-lo al correu quan es podria fer directament). Cal dir que aquest mòdul treballarà amb tots els *inputs* de l'aplicació (diferents assignatures, no una sola), per tant es va decidir que es treballaria amb llistes i s'analitzarien un per un els diferents elements. Per tal d'explicar d'una forma senzilla aquest mòdul, s'explicaran els elements de la llista que composarien una sessió d'una pràctica i aquelles variables que es necessitaran:

- Classe. Aula on es realitzarà la pràctica.
- Inici. Hora d'inici de la pràctica de laboratori. Format: "Hora:Minut:Segons"
- Final. Hora de finalització de la pràctica de laboratori. Format: "Hora:Minut:Segons"
- Dia. Dia en el qual es realitzarà la pràctica de laboratori. Format: "Dia-Mes-Any"
- Assignatura. Assignatura a la qual correspon la sessió de laboratori.
- Trigger. Amb quina anterioritat es desitja rebre la notificació de pràctica
- E-mail. A quin correu es vol rebre l'arxiu contenidor de pràctiques.

¹ L'explicació del funcionament del *Kivy language* es troba al apartat 5 de la memòria

Els mòduls que utilitza `arxicsmulti.py` i que són necessaris pel seu funcionament són els següents:

- Calendar del mòdul `iCalendar`. Permet crear el calendari global on s'integraran totes les pràctiques.
- `vText` del mòdul `iCalendar`. Permet integrar text als diferents camps de text que componen un esdeveniment del calendari (per exemple la localització).
- `datetime` del mòdul `datetime`. A partir d'uns *inputs* com són any, més, dia , hora, minut i zona horària permet integrar la hora d'inici i finalització dels esdeveniments.
- `timedelta` del mòdul `datetime`. Integra el concepte *trigger* a l'alarma de l'esdeveniment. El trigger és l'anticipació de l'alarma respecte l'esdeveniment al que correspon.
- `ICalendar`. Mòdul que permetrà la creació d'un esdeveniment diferent en cada iteració de les llistes amb la crida de `icalendar.Event()`.
- `smtplib`. Aquest mòdul permet l'obertura del servidor per enviar el correu electrònic.
- `MIMEMultipart` del mòdul `email.MIMEMultipart`. S'encarrega de donar un caràcter predefinit al missatge. Enviar arxius ics és `MIMEMultipart('alternative')`.
- `MIMEText` del mòdul `email.MIMEText`. Permet adjuntar text al cos del correu i donar el format desitjat. En el cas d'ETSEIB Calendar el format utilitzat és 'html'.
- `MIMEBase` del mòdul `email.MIMEBase`. Dona un caràcter o un altre a l'arxiu que s'adjunta al correu electrònic. L'arxiu .ics que s'enviarà a través del correu electrònic correspon a 'application'→'octet-stream' com es veurà al codi.
- Encoders del mòdul `email`. Permet codificar el correu electrònic per tal de tramitar-lo correctament.

4.1.1. Funcions arxicsmulti.py

- `calendar_creator(classe,inici,final,dia,assignatura,trigger,mail)`

Aquesta funció s'encarrega d'enviar el correu electrònic a la direcció de correu electrònic inserida a la pantalla de presentació de l'aplicació ETSEIB Calendar. A través dels mòduls que s'han explicat amb anterioritat i dels *inputs* en forma de llista per a les variables *classe*, *inici*, *final*, *dia*, *assignatura* i en forma de *string* per a *trigger* i *mail*. Cal dir que aquesta funció primerament realitzarà una iteració entorn els diferents elements de les llistes (totes tenen la mateixa longitud, ja que prèviament es creen tenint en compte el seu posterior ús). A partir de les dades i un text pre-definit pel correu que s'enviarà, s'adjunta l'arxiu .ics amb totes les pràctiques emmagatzemades.

La totalitat del *script* del de l'arxiu *arxicsmulti.py* es pot veure a l'annex C.

4.2. Filtro.py

El mòdul *filtro.py* s'ocupa d'accedir a la base de dades de les pràctiques de laboratori de l'ETSEIB i donar com a sortida diferents paràmetres. Consta de 3 funcions i cadascuna s'ocupa d'una determinada tasca. Cal tenir en compte que al cridar a un servidor de la xarxa cal tenir connexió a aquesta, ja que en cas contrari, Python no podrà executar la crida de variables i apareixeran errors. Això caldrà tenir-ho en compte pel la sèrie d'errors dels que cal protegir l'aplicació. Els mòduls de Python que caldrà importar per realitzar totes aquestes tasques són els següents:

- `json`. Permetrà llegir les dades de la base de dades de l'ETSEIB, les quals com ja s'ha parlat estaran en format json.
- `urllib`. Aquest mòdul permet obrir direccions web. En el nostre cas s'encarregarà d'accedir a la direcció que conté el document json que permet accedir als horaris de pràctiques.
- `HTMLParser`. S'encarrega de transformar el format html a format 'utf-8'

4.2.1. Funcions de Filtro.py

- `cerca(codi)`

Permet que a partir d'un codi d'assignatura es retorni una llista amb tots els grups que conté aquesta assignatura. A través de la crida de la base de dades de les pràctiques i la inserció d'un codi d'assignatura vàlid s'obté el seguit de grups que té aquesta assignatura, permetent saber si l'usuari de l'aplicació ha inserit o no un grup existent per l'assignatura seleccionada.

- `dic_codis_signatures(n)`

A partir d'un codi que pot ser "0", "1" o "2" es retornarà un diccionari amb tots els codis d'assignatura com a claus i tots els noms de les assignatures com a valors. El valor "0" correspondrà al bloc del GETI, "1" a GEQ i "2" a GEM. De cara a facilitar la escriptura dels *strings* aquest diccionari resultarà de gran utilitat, ja que únicament amb el codi de l'assignatura es podria aconseguir el nom complet en català.

- `cerca_horaris(codis,grups,trigger,mail)`

Funció més complexa del mòdul, a partir d'una llista de codis d'assignatures seleccionats, una altre de grups seleccionats, del *trigger* desitjat per els avisos i el *email* on es vol enviar el correu, obtindrà totes les dades de les assignatures i les inserirà a la funció `arxismulti.py` com a *inputs*. Simplement fa de pont entre l'aplicació i el creador de calendaris, donant com a resultat el correu electrònic amb l'arxiu calendari que rebrà l'usuari de l'aplicació.

La totalitat del *script* del de l'arxiu *filtro.py* es pot veure a l'annex D.

4.3. Main.py

Mòdul que s'encarrega de donar interfície gràfica a l'aplicació i que permet inserir tots els *inputs* necessaris per tal d'aconseguir l'objectiu final: el calendari personalitzat de pràctiques. Aquesta part del desenvolupament de l'aplicació és la que comporta més temps, ja que el fet de implementar la interfície gràfica és allò que comporta major temps i esforç normalment. Utilitzant paral·lelament el kv Language i el codi de Python així dit, permetrà dotar a l'aplicació de caràcter propi. Està formada per diferents classes, les quals permetran combinades crear les diferents pantalles i funcions de l'aplicació. Degut a la complexitat d'aquest mòdul s'explicaran les classes que el componen i posteriorment en un següent apartat s'entrarà en les funcions que corresponen a cada classe. Els mòduls que cal importar són el següents:

- App del mòdul `kivy.app`. Permetrà executar la Kivy app i que s'iniciï el bucle que donarà

com a resultat la interfície gràfica de l'aplicació. Base per crear les Kivy app.

- Builder del mòdul `kivy.lang`. Permetrà utilitzar el Kivy *language* com a llenguatge per crear els elements de les diferents pantalles, facilitant la creació de elements com ara botons, entrades de text o imatges, que en cas contrari s'hauria de fer com a `pygame`.
- ScreenManager del mòdul `kivy.uix.screenmanager`. Permetrà la gestió de les diferents pantalles i el seu enllaç a través de diferents transicions.
- Screen del mòdul `kivy.uix.screenmanager`. Permetrà la creació de l'element més bàsic de l'aplicació, una pantalla.
- SwapTransition del mòdul `kivy.uix.screenmanager`. Permetrà modificar el tipus de transició predeterminada de l'aplicació a SwapTransition (fusió d'una pantalla a la següent).
- TextInput del mòdul `kivy.uix.textinput`. Permetrà la creació d'un *widget* que faciliti l'entrada de text per l'usuari de l'aplicació.
- Label del mòdul `kivy.uix.label`. Permetrà la creació d'una etiqueta de text.
- GridLayout del mòdul `kivy.uix.gridlayout`. Permetrà la creació d'una pantalla en forma de *grid* (quadícula) . El fet de que es faci una quadricula permetrà una organització proporcional en l'entorn de la pantalla.
- Button del mòdul `kivy.uix.button`. Permetrà la creació d'un botó.
- DictProperty del mòdul `kivy.properties`. Permetrà la creació d'un diccionari que podrà ser modificat des de qualsevol classe.
- StringProperty. Permetrà la creació d'una cadena de text que podrà ser modificada des de qualsevol classe.
- SMTPRecipientsRefused del mòdul `smtplib`. Permetrà detectar si apareix un error a la hora d'enviar el correu electrònic, com en el cas de no introduir una direcció correcta.
- Window del mòdul `kivy.core.window`. Permetrà calcular les mesures de la pantalla on s'accionarà l'aplicació.
- Popup del mòdul `kivy.uix.popup`. Permetrà la creació d'un *popup* que notifiqui que un procés no s'està fent correctament o que ja realitzat el procés.

Tots aquests mòduls permetran crear en la seva totalitat les següents classes:

4.3.1. Classe Welcome

Hereta del mòdul *Screen* de Kivy i permetrà crear la pantalla inicial o s'inseriran el correu electrònic i l'anterioritat amb la que es vol rebre la notificació per correu electrònic. La totalitat de l'estructuració dels diferents *widgets* es realitza a través del *kv language*, tal i com es veurà en apartats següents de la memòria. Aquesta classe té integrades les següents funcions:

- *emmagatzema_mail(self, text_inserit)*. Emmagatzema els textinputs que s'inseriran a l'entorn de l'aplicació. En cas de no insereix res tindrà un valor per defecte ("default"). L'objectiu d'aquesta aplicació és emmagatzemar el correu electrònic de l'usuari.
- *info(self)*. Emmagatzema informació de l'aplicació, s'encarrega d'obrir un *popup* amb informació sobre l'entorn de l'aplicació i l'autor. Aquesta funció s'accionarà quan és polsi el botó "i" de la pantalla inicial.
- *emmagatzetma_trigger(self, seleccio)*. Permet emmagatzemar la selecció realitzada al *spinner*. En cas de no seleccionar cap valor prendrà un valor per defecte ("default"). Aquesta funció s'accionarà al seleccionar un valor del *spinner*, que indicarà amb quina anterioritat es realitzarà l'avís de la pràctica.
- *close(self)*. Permet tancar l'aplicació quan es premi el botó "Surt".
- *ready(self)*. Permet realitzar una comprovació abans d'avançar a la següent pantalla (selecció d'assignatures 1). Aquesta comprovació es realitzarà en 3 variables: self.a.mail (correu electrònic emmagatzemat al *textinput*), self.a.trigger (anterioritat de l'avís emmagatzemat al *spinner*) i internet (variables que permet determinar si hi ha connexió a xarxa o no). En cas de no inserir valors al *textinput* i al *spinner*, o bé de no disposar de connexió a internet l'aplicació advertirà de completar tots els camps i no permetrà avançar a la següent pantalla. En cas de que es compleixin les 3 premisses s'avançarà a la següent pantalla, la classe MainScreen.

4.3.2. Classe MainScreen

Hereta del mòdul *Screen* de Kivy i permetrà crear la pantalla amb les *checkboxes* per les primeres 7 assignatures. A més a més disposarà de dos botons que permetran avançar a la següent pantalla o bé retornar a la pantalla inicial. Les funcions que integren aquesta classe són les següents:

- *emmagatzema(self, codi, value)*. Emmagatzema els valors de les caselles dels *checkbox* de les diferents assignatures. A través d'un diccionari es podrà controlar de quines assignatures es vol obtenir informació de les pràctiques i de quines no. Inicialment el diccionari dels codis disposarà de tots els codis de les assignatures amb

pràctiques de laboratori amb el valor “False” (no seleccionades). En cas de seleccionar un *checkbox* canviarà el valor a “True”.

4.3.3. Classe SecondScreen

Hereta del mòdul *Screen* de Kivy i permetrà crear la pantalla amb les *checkbox* per les altres 6 assignatures. A més a més disposarà de dos botons que permetran avançar a la següent o bé retornar a la primera pantalla de selecció d'assignatures. Funcions:

- *emmagatzema(self, codi, value)*. Emmagatzema els valors de les caselles dels *checkbox* de les diferents assignatures. A través d'un diccionari es podrà controlar de quines assignatures es vol obtenir informació de les pràctiques i de quines no. Inicialment el diccionari dels codis disposarà de tots els codis de les assignatures amb pràctiques de laboratori amb el valor “False” (no seleccionades). En cas de seleccionar un *checkbox* canviarà el valor a “True”.
- *ready(self)*. Filtre previ al pas a la següent pantalla. En cas de no seleccionar cap *checkbox* en les 2 pantalles avisarà a l'usuari de la necessitat de fer-ho, ja que en cas contrari no podrà avançar a la següent pantalla. A través d'un recorregut de tots els valors del diccionari amb les seleccions de les *checkbox* s'estimarà si es pot avançar a la pantalla de selecció de grups. A més a més es tornarà a revisar la connexió a internet, ja que per tal de crear la següent pantalla serà necessària connexió a internet. En cas de no disposar de connexió a la xarxa s'obrirà un *popup* avisant a l'usuari.

4.3.4. Classe AnotherScreen

Hereta del mòdul *Screen* de Kivy i permetrà crear la pantalla a partir de les seleccions prèvies de les *checkbox*. En aquesta pantalla es podran introduir els grups als que l'alumne està matriculat. A més a més disposarà de dos botons que permetran executar la creació del calendari i enviar el correu o bé retornar a la pantalla anterior. Aquesta classe integra les següents funcions:

- *on_pre_enter(self, *args)*. A partir de les seleccions prèvies dels *checkbox* de les assignatures es definirà aquesta pantalla, que crearà uns *textinput* on l'estudiant podrà inserir els grups de les assignatures que ha seleccionat prèviament. Aquests *textinputs* mostraran les possibilitats de grups que té l'estudiant per aquella assignatura. Les entrades de text que es facin als *textinput* seran emmagatzemades a través d'una altre funció.
- *emmagatzema_textinputs(self,p)*. Emmagatzema els *textinput* que es realitzin en un diccionari, on la clau serà el codi de l'assignatura i el valor el valor inserit en el *textinput*.

- *executa(self)*. Funció que a partir dels *texinput* aconseguirà els arxius necessaris per la creació de l'arxiu *iCalendar*. Aquesta funció treballarà bàsicament amb els codis d'assignatura i els *textinputs* inserits a la pantalla. A través d'una primera validació dels valors inserits (cal que siguin uns valors vàlids per passar al següent pas de validació) es farà una crida al mòdul *filtro.py* que permetrà accedir al mòdul *arxicsmulti.py*, que crearà l'arxiu ics i intentarà enviar el correu electrònic al correu que l'usuari havia inserits a la pantalla inicial. En cas de que aquest correu electrònic no sigui vàlid (compte de correu electrònic inexistent) es forçarà un error per tal d'evitar que l'aplicació es tanqui de manera sobtada, i es mostrarà un *popup* amb un text informant que el correu electrònic inserit no es vàlid, juntament d'un botó que permetrà anar a la primera pantalla a inserir un de vàlid. Si aquestes dos premisses es compleixen s'enviarà el correu electrònic amb el calendari de les pràctiques i s'obrirà un *popup* indicant que el procés s'ha realitzat correctament.
- *change1(Screen)*. Aquesta funció permetrà canviar a la primera pantalla per inserir de nou el correu electrònic. Només apareixerà la possibilitat d'accionar aquesta funció quan s'obri el *popup* de correu electrònic no vàlid.
- *change2(Screen)*. Permet retrocedir a la segona pantalla de selecció d'assignatures, ja que existeix la possibilitat de que l'usuari pugui haver oblidat alguna selecció.

4.3.5. Classe Root

Hereta del mòdul *ScreenManager* de Kivy i permetrà crear un arxiu contenidor on definir de forma més senzilla tots els elements de cada pantalla. Al següent punt de la memòria s'explicarà de forma senzilla com s'obté cada element de forma més senzilla amb Kivy. No integra cap funció.

4.3.6. Classe SimpleKivy

Hereta del mòdul *App* de Kivy i permetrà donar-li format d'aplicació al nostre codi. Emmagatzemarà les variables globals de l'aplicació: *textinputs* de correu electrònic, *trigger* de l'avís i seleccions de les *checkbox*. Integra la següent funció:

- *build(self)*. Retorna directament la classe Root i permet iniciar el bucle de l'aplicació. A través de la classe Root es podran definir amb més facilitat l'entorn de l'aplicació.

La totalitat del *script* del de l'arxiu *main.py* es pot veure a l'annex E.

4.4. Taula resum funcions i arxius

Arxicsmulti.py:

Funció	Descripció
calendar_creator	A partir de les dades tractades pel mòdul filtro.py permet crear l'arxiu contenidor iCalendar i l'envia per correu electrònic al email seleccionat a la plataforma ETSEIB Calendar

Taula 4.1: Resum de les funcions de l'arxiu arxicsmulti.py

Filtro.py:

Funció	Descripció
cerca	A partir d'un codi d'assignatura s'obtenen tots els grups de pràctiques existents
dic_codis_assignatures	Retorna un diccionari amb tots els codis d'assignatura i el nom de la assignatura
cerca_horaris	Aportant dos llistes (codis d'assignatura i grups) ens retorna un document amb totes les practiques integrades

Taula 4.2: Resum de les funcions de l'arxiu filtro.py

Main.py:

Classe	Nivell 1 Funció	Nivell 2 Funció	Descripció
Welcome	__init__	-	Inicialitza les variables globals de l'aplicació
Welcome	emmagatzema_email	-	Emmagatzema les entrades de text de correu
Welcome	info	-	Mostra informació de l'aplicació a través d'un popup
Welcome	emmagatzema_trigger	-	Emmagatzema l'antelació amb la que es durà a terme l'avís per correu
Welcome	close	-	Permet tancar l'aplicació
Welcome	ready	-	Comprovació dels camps inserits i de la connexió a la xarxa i en cas d'estar tot correcte avança a la següent pantalla
MainScreen	__init__	-	Inicialitza les variables globals de l'aplicació
MainScreen	emmagatzema	-	Captura els valors de les checkboxes de la primera pantalla de selecció d'assignatures i les guarda dins d'un diccionari que apareix com a variable global (big_dict)
SecondScreen	__init__	-	Inicialitza les variables globals de l'aplicació
SecondScreen	emmagatzema	-	Captura els valors de les checkboxes de la segona pantalla de selecció d'assignatures i les guarda dins d'un diccionari que apareix com a variable global (big_dict)

SecondScreen	ready	-	Comprovació de la connexió a la xarxa i dels valors de les checkboxes (alguna ha d'ésser seleccionada) i en cas de ser vàlida la selecció permetrà avançar a la pantalla de selecció de grups
AnotherScreen	on_pre_enter	-	Crea la pantalla de selecció de grups a partir de les seleccions d'assignatures realitzada prèviament
AnotherScreen	on_pre_enter	emmagatzema_textinputs	Emmagatzema els textinputs de cada casella creada automàticament al inicialitzar la classe
AnotherScreen	change1	-	Permet passar a la pantalla inicial
AnotherScreen	change2	-	Permet passar a la pantalla de selecció d'assignatures
AnotherScreen	executa	-	A partir dels <i>inputs</i> comprovarà si són vàlids i obtindrà l'arxiu contenidor a través d'altres mòduls
AnotherScreen	executa	popitup	Funció que s'encarrega d'indicar que s'ha realitzat l'operació correctament

Taula 4.3: Resum de les classes i funcions de l'arxiu main.py

5. Kivy Language

Una part molt important del desenvolupament d'ETSEIB Calendar va ser decidir amb quina plataforma es desenvoluparia. Tal i com es va comentar als primers punts de la memòria, Kivy aportava facilitats a la creació d'elements, com ara botons, entrades de text o selectors. En aquest punt de la memòria s'explicarà el funcionament de l'anomenat com Kivy *language*, una de les parts més importants en l'arquitectura de l'aplicatiu.

Lo primer que cal entendre és que cada classe que es crea al document de Python que executarà el programari Kivy esdevindrà una pantalla de l'aplicació. A aquestes classes caldrà aportar a més a més si es desitja crear una classe contenidora de la resta (per crear la classe a través de Kivy *language*) i una classe que inici el bucle de l'aplicació (en el nostre cas és la classe *SimpleKivy* que hereta directament del mòdul *App* de Kivy). La classe contenidora no és obligatòria però tenint en compte que facilita l'organització i creació d'elements no seria intel·ligent prescindir d'ella. En el cas de l'aplicació desenvolupada en el projecte, la classe s'anomena *Root* i hereta del mòdul *ScreenManager* que permetrà poder facilitar la creació de pantalles.

Per tal d'iniciar la construcció a partir de Kivy *language* caldrà inserir aquesta línia al codi:

Builder.load_string(arquitectura diferents pantalles). Així es definiren les diferents classes i a més a més es podrà compatibilitzar l'ús del codi de Python en la creació d'elements. Les primeres línies del codi consistiran en la definició d'elements com per exemple la transició entre pantalla i pantalla:

```
#:import FadeTransition kivy.uix.screenmanager.FadeTransition
```

Després es pot procedir a definir com serà l'aplicació, primer definint quines pantalles contindrà la classe contenidora *Root*:

<Root>:

```
transition: FadeTransition()
```

```
Welcome:
```

```
name: 'welcome'
```

```
MainScreen:
```

```
name: 'main'
```


SecondScreen:

name: 'second'

AnotherScreen:

name: 'another'

Bàsicament es defineixen les diferents pantalles que tindrà l'aplicació i se'ls atorga amb un nom que les denominarà durant tota l'aplicació i permetrà realitzar les transicions a altres pantalles. Un altre pas previ a la definició de les diferents pantalles és la definició o modificació de propietats per a tots els elements d'una mateixa "família". En el cas d'ETSEIB Calendar es van modificar les següents propietats:

<Image>

keep_ratio: True

allow_stretch: True

opacity: 1

<Popup>

size_hint: 0.7, 0.35

background_color: 0, 0, 0, 0

<CheckBox>

background_checkbox_normal: 'unchecked.png'

background_checkbox_down: 'checked.png'

<Label>

markup: True

font_size: sp(16)

<TextInput>

padding: [12,(self.height - self.line_height) / 2]

L'objectiu d'aquestes modificacions són els següents:

- Imatge. Que es mantingui la relació entre distàncies quan s'insereixi la imatge a qualsevol pantalla i sigui d'opacitat 100% (sense transparència).
- Popup. Que ocupes un 70% de la pantalla en l'eix X i un 35% en l'eix Y. A més a més que el fons del Popup fos transparent i no alteres en gran mesura la pantalla de l'aplicació.
- CheckBox. Modificació de les imatges que s'utilitzen de forma predefinida. A canvi es van crear unes altres que van permetre una millor visualització de l'aplicació.
- Label. Alçada de la font utilitzada fixada de forma proporcional a la pantalla amb "sp" i a més a més s'activa el poder utilitzar caràcters especials amb el *markup*.
- TextInput. L'alçada a la que estaria el text respecte el quadre de text seria la mateixa per dalt i per baix, restant el text centrat respecte el quadre de text.

Un cop modificades de forma general les propietats de diferents elements com són els mencionats anteriorment, es pot iniciar la creació de les pantalles de l'aplicació:

<Welcome>:

canvas.before:

Color:

rgba: 0.5, 0.5, 0.5, 0.8

BorderImage:

border: 10, 10, 10, 10

source: 'etseib.jpg'

pos: self.pos

size: self.size

Amb aquestes línies de codi es defineix el fons de pantalla que tindrà la pantalla de benvinguda (classe *Welcome*). A més a més es poden modificar propietats com ara la transparència, el color de la màscara o la proporció de la pantalla que es vol cobrir.

FloatLayout:

De totes les possibilitats de *layouts* disponibles a Kivy es va decidir escollir *FloatLayout* per la

facilitat d'incorporació d'elements a unes coordenades determinades de la pantalla. Aquesta *layout* serà utilitzada únicament en aquesta pantalla, i constarà del següents elements:

Image:

```
source: 'escriucorreu.png'

size_hint_x: 0.16

pos_hint: {'right':0.395,'center_y':0.6}
```

Aquesta imatge ocuparà una proporció del 16% de la pantalla en l'eix X i s'adaptarà a totes les pantalles on s'inicialitzi l'aplicació. A més a més tindrà el seu extrem dret a la coordenada 39,5% de l'amplada i l'alçada 60%. La raó per la utilització d'una imatge que indiqui que insereixis el correu i no una *Label* és per el fet d'adaptar-la de forma proporcional a l'eix X automàticament, cosa que no es pot fer amb una *Label* i que provocaria una visualització on els elements es solaparien entre ells.

Image:

```
source: 'best.png'

size_hint: 0.85

pos_hint: {'center_x':0.5,'center_y':0.375}
```

Mateixes condicions que la imatge anterior, però en altres coordenades de X i de Y (50% amplada i 37,5% d'alçada i ocuparà un 85% de l'amplada de la pantalla.

Spinner:

```
text: "Quan vols rebre l'avís?"

values:('24 hores abans','8 hores abans','2 hores abans','1 hora
abans')

pos_hint: {'center_x':0.5,'center_y':0.19}

size_hint: 0.2, 0.0575

on_text: root.emmagatzema_trigger(self.text)
```

Un *spinner* és un selector que permet escollir entre diferents opcions. En aquest cas cal modificar únicament les coordenades on es voldrà inserir l'element (50% X i 19% Y) i les

dimensions respecte la pantalla (20% de l'amplada total de la pantalla i un 5,75% de l'alçada). També s'incorpora una funció que s'executarà quan es seleccioni una possibilitat i que emmagatzemarà la selecció a través de la funció *root.emmagatzema_trigger(self.text)*.

Button:

```
text: "[b]Surt[/b]"

background_color: 0.34, 1.83, 2, 1

on_release: root.close()

size_hint: 0.5, 0.1111111111

pos_hint: {'center_x':0.25,'center_y':0.06}
```

Creació d'un botó que ocuparà el 50% de la pantalla en l'eix X i un 1/9 de l'eix Y. Esta situat a la part inferior esquerra i quan sigui premut permetrà sortir de l'aplicació. A més a més es defineixen propietats com ara el color del botó i el text que es desitja.

Button:

```
text: '[b]Següent[/b]'

background_color: 6,1.5,0,1

on_release: root.ready()

size_hint: 0.5, 0.1111111111

pos_hint: {'center_x':0.75,'center_y': 0.06}
```

Creació d'un botó que ocuparà el 50% de la pantalla en l'eix X i un 1/9 de l'eix Y. Esta situat a la part inferior dreta i quan sigui premut permetrà executar la funció *ready* que permetrà revisar si s'han inserit correctament els camps i en cas afirmatiu avançarà a la següent pantalla. A més a més es defineixen propietats com ara el color del botó i el text que es desitja.

TextInput:

```
multiline: False

input_type: 'mail'

size_hint: 0.2, 0.0595
```

```
pos_hint: {'center_x':0.5,'center_y':0.6}

hint_text:'exemple@gmail.com'

on_text: root.emmagatzema_email(self.text)
```

Aquest codi permet la creació d'una entrada de text, que ocuparà un 20% de l'amplada de la pantalla i un 5,95% de l'alçada. Estarà situat a la meitat de l'eix X i a un 60% de l'alçada. A més a més tindrà un missatge predefinit (exemple@gmail.com) i quan s'escrigui qualsevol caràcter s'executarà la funció *emmagatzema_email*, que canviarà el valor de la variable global *email* que es emmagatzema a la classe *SimpleKivy*.

Image:

```
source: 'calendar.png'

size_hint: 0.52, 0.375

pos_hint: {'center_x': 0.5, 'center_y': 0.83}
```

Inserció de la imatge del títol de l'aplicació. Ocupa un 52% de l'amplada de la pantalla i un 37,5% de l'alçada. Esta centrada en l'eix X i a un 83% de l'eix Y.

Button:

```
background_down: 'help2.png'

background_normal: 'help.png'

size_hint: 0.1,0.15

pos_hint: {'center_x': 0.85, 'center_y': 0.85}

on_release: root.info()
```

Creació d'un botó amb imatge personalitzades quan esta premut o no. Aquestes imatges s'han creat de forma paral·lela per canviar el format del botó a un altre més visual. ocupa un 10% de l'eix X i està situat a un 85% de l'amplada i un 85% de l'alçada. En cas de ser accionat s'obrirà un *popup* amb informació.

Amb tots aquest elements la primera pantalla de presentació queda definida, però cal seguir definint la resta de pantalles. La següent pantalla és:

<MainScreen>:

```
canvas.before:
```

```
    Color:
```

```
        rgba: 1, 1, 1, 0.8
```

```
    BorderImage:
```

```
        border: 10, 10, 10, 10
```

```
        source: 'hd.jpg'
```

```
        pos: self.pos
```

```
        size: self.size
```

MainScreen al igual que *Welcome* disposa d'una imatge de fons que s'adapta a la totalitat de la pantalla. Disposa a més a més dels següents elements:

```
GridLayout:
```

```
    cols: 2
```

A diferència de la primera pantalla, aquesta disposa de la denominada com *GridLayout*, una estructura de pantalla que es basa en afegir els *widgets* en forma de quadricula. Per realitzar aquesta quadricula es poden fixar les files o les columnes, en aquest cas s'han fixat les columnes a dos.

```
    Label:
```

```
        text: "[b]Selecciona Assignatures (1/2)[/b]"
```

```
        font_size: sp(17)
```

```
    Label:
```

```
        text: " "
```

Aquestes dos *labels* permetran tenir una capçalera a la pantalla, amb una font més gran respecte la mida normal que s'havia definit de manera global (era 16 i aquesta és 17), i a més a més en negreta. La segon *label* estarà en blanc i deixarà un buit a la part dreta de la pantalla. A partir d'aquest moment la pantalla disposarà de 7 files on totes seguiran el mateix patró, primer un *checkbox* i després una *label* amb el nom de l'assignatura:

CheckBox:

```
on_active:root.emmagatzema('240031',self.active)
```

Label:

```
text: "[b]Electromagnetisme[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240033',self.active)
```

Label:

```
text: "[b]Materials[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240052',self.active)
```

Label:

```
text: "[b]Termodinàmica[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240053',self.active)
```

Label:

```
text: "[b]Electrotècnia[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240054',self.active)
```

Label:

```
text: "[b]Mecànica dels Medis Continus[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240061',self.active)
```

Label:

```
text: "[b]Mecànica de Fluids[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240063',self.active)
```

Label:

```
text: "[b]Resistència de Materials[/b]"
```

En cas de seleccionar un *checkbox* s'accionarà una funció del document Python que permetrà l'emmagatzemament d'una variable global en un diccionari (*big_dict*). A la última fila es troben els següents elements:

Button:

```
text: "[b]Inici[/b]"
```

```
background_color: 0.34, 1.83, 2, 1
```

```
on_release:
```

```
root.parent.current='welcome'
```

Un botó que ocuparà el 50% de l'amplada de la pantalla i que permetrà anar a la pantalla inicial. Cal tenir en compte que com la pantalla es proporcional a nivell de files totes ocuparen una novena part de l'alçada total. Per últim trobem:

Button:

```
text: "[b]Següent[/b]"
```

```
background_color: 6,1.5,0,1
```

```
on_release:
```

```
root.parent.current='second'
```

Botó que permet el canvi a la segona pantalla de selecció d'assignatures, l'anomenada com 'second' al principi del document de Kivy *language*. Aquesta *SecondScreen* tindrà els següents elements:

<SecondScreen>:

```
canvas.before:
```


Color:

```
rgba: 1, 1, 1, 0.7
```

BorderImage:

```
border: 10, 10, 10, 10
```

```
source: 'hd.jpg'
```

```
pos: self.pos
```

```
size: self.size
```

Aquesta segona pantalla de selecció d'assignatures té la mateixa disposició d'elements que la pantalla anterior, mateix fons de pantalla, i mateix format de *layout*, en forma de graella. A la primera pantalla de selecció d'assignatures es trobaven les primeres 7 assignatures (la pantalla quedada dividida en 9 files). En aquesta segona es troben les altres 6 assignatures i per tal de mantenir el mateix format i espais s'integra una fila sencera de labels buides (ocupen l'espai i passa la pantalla de 8 a 9 files).

GridLayout:

```
cols: 2
```

Label:

```
text: "[b]Selecciona Assignatures (2/2)[/b]"
```

```
font_size: sp(17)
```

Label:

```
text: " "
```

CheckBox:

```
on_active:root.emmagatzema('240072',self.active)
```

Label:

```
text: "[b]Electrònica[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240073',self.active)
```

Label:

```
text: "[b]Sistemes de Fabricació[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240151',self.active)
```

Label:

```
text: "[b]Tecnologia i Selecció de Materials[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240161',self.active)
```

Label:

```
text: "[b]Màquines Elèctriques[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240171',self.active)
```

Label:

```
text: "[b]Termotècnia[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240172',self.active)
```

Label:

```
text: "[b]Control Automàtic[/b]"
```

Label:

```
text: ""
```

Label:

```
text: ""
```

Button:

```
text: "[b]Anterior[/b]"

background_color: 0.34, 1.83, 2, 1

on_release:

    root.parent.transition.direction = 'right'

    root.parent.current='main'
```

Button:

```
text: "[b]Següent[/b]"

background_color: 6,1.5,0,1

on_release:

    root.ready()
```

A la última fila trobem dos botons que permeten retrocedir a la pantalla anterior i avançar a la pantalla de selecció de grups (executant la funció *ready*). La funció *ready* consisteix en verificar que s'ha seleccionat alguna assignatura i si hi ha connexió a internet ja que en cas de no disposar de connexió no es podrà accedir a la base de dades de les pràctiques de laboratori. Per últim es defineix la classe *AnotherScreen*:

<AnotherScreen>:

canvas.before:

Color:

```
rgba: 1, 1, 1, 0.8
```

BorderImage:

```
border: 10, 10, 10, 10

source: 'hd.jpg'

pos: self.pos

size: self.size
```

Mateix fons que les pantalles anteriors de selecció d'assignatures, aquesta classe es caracteritza per ser una pantalla que es genera a partir de les seleccions de les pantalles anteriors. Per cada assignatura seleccionada s'inserirà una fila més, amb una *label* per indicar el nom de l'assignatura i un *textinput* per escriure els grups.

GridLayout:

```
id: container
```

```
cols: 2
```

Al contrari que les altres classes, *AnotherScreen* no té elements creats en Kivy *language*, ja que al crear-se a partir d'un iteració en el llenguatge Python es millor fer-ho en llenguatge Python, encara que això comporti escriure més línies de codi. A través de la definició de propietats com el nombre de columnes (2) i un nom identificador de la pantalla ("*container*") podrem interactuar amb el Kivy *language*.

La totalitat del codi de Kivy *language* amb el que s'obté com a resultat ETSEIB Calendar es pot trobar a l'annex F.

6. Creació de la pantalla de selecció de grups

Segurament una de les tasques més complicades en el desenvolupament d'ETSEIB Calendar va ser el desenvolupament d'una pantalla que es crees a partir de les seleccions d'altres pantalles, concretament de les pantalles *MainScreen* i *SecondScreen*.

El principi teòric és senzill: l'usuari, per exemple, selecciona 4 assignatures de la primera pantalla de selecció i 1 de la segona, amb lo que s'esperaria que a la pantalla d'inserció de grups apareguessin 5 *textinputs* que permetessin inserir quins grups desitges seleccionar. El problema apareix quan es vol desenvolupar en Kivy *language*, ja que fer iteracions d'un element emmagatzemat en una variable per crear elements a Kivy *language* es força complexa. La solució ve detallada en aquest punt, on en comptes de treballar en Kivy *language* per la creació d'elements (sempre és més recomanable per facilitar les tasques), es creen els elements a partir de codi Python. Tal i com s'ha parlat al punt anterior, la classe *AnotherScreen* està identificada amb el *id* "container". A partir de la crida d'aquest *id* es podran crear elements interactuant en ambdós llenguatges. El codi que cal per poder executar la creació d'una pantalla en funció de les seleccions és el següent:

```

1. class AnotherScreen(Screen):
2.     def on_pre_enter(self, *args):
3.         self.ids.container.clear_widgets()
4.         dic_assignatures=filtro.dic_codis_assignatures(0)
5.         a = App.get_running_app()
6.         text_inputs={}
7.         keys=[]
8.         self.ids.container.add_widget(Label(text="[b]Assignatura[/b]",font_size='
17sp'))
9.         self.ids.container.add_widget(Label(text="[b]Escriu el grup[/b]",font_siz
e='17sp'))
10.        def emmagatzema_textinputs(self,seleccio):
11.            text_inputs[self.id]=seleccio
12.            for e in a.big_dict:
13.                keys.append(e)
14.                keys.sort()
15.                for codi in keys:
16.                    if a.big_dict[codi]==True:
17.                        self.ids.container.add_widget(Label(text='[b]'+codi+'-
'+dic_assignatures[codi].encode("utf-8")+'[b]'))
18.                        suggeriment_grups=str(filtro.cerca(codi.encode("utf-8")))[1:-
1].replace("'", ""))
19.                        self.codi=TextInput(hint_text=suggeriment_grups,multiline=False,i
d=codi,write_tab=False,input_type='number',input_filter='int',background_color=[1,1,1
,0.8],font_size='15sp')
20.                        self.codi.bind(text=emmagatzema_textinputs)
21.                        self.ids.container.add_widget(self.codi)
22.        def change1(Screen):
23.            self.parent.current='welcome'
24.        def change2(Screen):
25.            self.parent.current='second'
26.        def executa(self):
27.            ass=[]
28.            grup=[]

```

```

29.         error=False
30.         def popitup():
31.             content=Label(text='[b]Operació realitzada amb èxit, revisa el teu email[/b]', font_size='17sp')
32.             popup = Popup(title='Operació realitzada',content=content)
33.             popup.open()
34.             content.bind(on_press=popup.dismiss)
35.             for e in text_inputs:
36.                 try:
37.                     if text_inputs[e] in filtro.cerca(e.encode("utf-8")):
38.                         x=float(text_inputs[e])
39.                         ass.append(e)
40.                         grup.append(text_inputs[e].encode("utf-8"))
41.                     else:
42.                         raise ValueError('No existeix la teva selecció')
43.                 except:
44.                     content=Label(text='[b]Insereix grups vàlids[/b]')
45.                     popup = Popup(title='Error de grup',content=content,size_hint=(0.7, 0.5), background_color=(0, 0, 0, 0),font_size='15sp')
46.                     popup.open()
47.                     content.bind(on_press=popup.dismiss)
48.                     ass=[]
49.                     grup=[]
50.                     error=True
51.                     break
52.             if error==False:
53.                 try:
54.                     return filtro.cerca_horaris(ass,grup,int(a.trigger),a.email),
popitup()
55.                 except SMTPRecipientsRefused:
56.                     content=Button(text='[b]El email inserit no és vàlid[/b]',on_release=change1 ,size_hint= (0.6, 0.3), font_size='15sp')
57.                     popup = Popup(title='Error de email',content=content)
58.                     popup.open()
59.                     content.bind(on_press=popup.dismiss)
60.                     b1=Button(text='[b]Anterior[/b]',background_color=[0.34, 1.83, 2, 1], size_hint_y= None, height= int(Window.height)/8.8 )
61.                     self.ids.container.add_widget(b1)
62.                     b2=Button(text='[b]Executa[/b]',background_color=[6,1.5,0,1],size_hint_y= None, height= int(Window.height)/8.8)
63.                     self.ids.container.add_widget(b2)
64.                     b1.bind(on_release=change2)
65.                     b2.bind(on_release=executa)

```

A continuació es comentarà les funcions que realitzen cadascuna de les funcions i línies d'aquest codi. La funció *on_pre_enter(self, *args)* s'encarregarà de crear automàticament el formulari on inserir els grups de les assignatures seleccionades prèviament.

La primera acció que farà serà esborrar la pantalla perquè en cas de tornar a la pantalla anterior i retornar a la pantalla *AnotherScreen* (línia 3). El següent pas consistirà en fer una crida al mòdul *filtro.py* per tal de facilitar la escriptura dels noms de les assignatures. Tal i com s'ha comentat anteriorment a la memòria, el mòdul *filtro.py* conté una funció que retorna un diccionari amb claus els codis de les assignatures i valors els *strings* que corresponen als noms en format *utf-8*. Aquest diccionari es nomenarà *dic_assignatures* (línia 4). A la línia 5

simplement es realitza una crida a les variables globals a través de l'execució de *App.get_running_app()*. Posteriorment s'inicialitza un diccionari on s'emmagatzemaran totes les entrades de text inserides en les diferents *textinputs* (línea 6) i una llista buida anomenada *keys* (línia 7). A les línies 8 i 9 s'insereixen dos *labels* a la classe *AnotherScreen*, que cal recordar que tenia format *GridLayout* (quadricula) . Les línies 10 i 11 es comentaran més endavant, ja que per executar funcions a Kivy cal primer definir-las i després realitzar la crida, en cas contrari dona error.

Un cop creada la base d'aquesta última pantalla, comença la iteració de la variable que emmagatzemava les assignatures seleccionades en pantalles anteriors. Primer de tot s'ordenaran totes les claus del diccionari clau que agrupa les seleccions, ja que en cas contrari sempre sortiria d'un ordre diferent la iteració. Això es realitza a la línia 12 i 13 (iteració) i el resultat final s'obté a la 14 (emmagatzemat dins de la variable *keys*). Amb totes les claus disponibles es comença la comprovació de la selecció a les pantalles anteriors (línia 15 fins línia 21). Iterant entorn al diccionari global (cal recordar que emmagatzemava els codis d'assignatura com a clau i un booleà que indicava la seva selecció com a valor), es comprovarà codi a codi si ha sigut seleccionada en les *checkboxes* anteriors, i en cas de que el valor sigui *True*, es crearà una nova fila a la pantalla (línia 16). A la línia 17 es crearà un *string* que consistirà en Codi-Nom assignatura i que serà afegit a la pantalla, ocupant la part esquerra de totes les files que siguin necessàries segons les seleccions d'assignatures de les pantalles anteriors. Un cop creat l'indicador de l'assignatura, es desenvoluparà l'anomenat com *suggestiment_grups* (línia 18) , un *string* que apareixerà al *textinput* quan no sigui seleccionat i que indicarà les possibilitats de grups de pràctiques per cada assignatura. A la línia 19 es crearà definitivament el *textinput*, que permetrà passar a la següent inserció de text (en cas d'existir) al primer tabulador, que únicament permetrà inserir números enters, tindrà el suggeriment esmentat anteriorment i a més a més farà que a Android aparegui el teclat numèric en comptes de l'alfanumèric. A la línia 20 es defineix la funció que caldrà executar cada cop que hi hagi una inserció de text (línies 10 i 11), que permetrà emmagatzemar en un diccionari la totalitat de les insercions aportades a l'aplicació i tindrà com a claus els codis de les assignatures. Finalment a la línia 21 s'insereix a la pantalla el *widget* de l'entrada de text i es torna a començar la iteració per la següent clau. Les línies de la 22 a la 25 corresponen a dos funcions que permetran canviar a la pantalla inicial (línia 22) o bé a l'anterior, la de selecció d'assignatures (línia 24). De la 26 a la 59 es defineix la funció executa de la que es parlarà més endavant, i que executarà la crida dels altres mòduls per aconseguir l'objectiu final d'ETSEIB Calendar, l'arxiu *iCalendar*. Finalment s'estableixen els dos botons, un que permet anar a la pantalla anterior (línia 60 i 61) i ocuparà un 11,11% de l'alçada de la pantalla (per tal de respectar dimensionament de l'aplicació) i un altre que cridarà la funció executa (línia 62 i 63). Les funcions que realitzen al ser premuts són a les línies 64 i 65.

Estudiant les línies 26 a 59 exhaustivament es pot veure tot l'arbre d'errors que té la l'aplicació

en quant es vol executar l'algoritme creador del calendari. A les línies 27 i 28 es creen dos llistes buides de assignatures i grups seleccionats, mentres que a la 29 es crea una variable booleana anomenada *error* que indicarà si s'ha produït algun tipus d'error. De la 30 a la 34 es defineix un missatge que sortirà quan es realitzi l'operació amb èxit, a través d'un *popup*. A la línia 35 s'inicia la iteració de les entrades de text per tal de comprovar si són vàlides i en cas de no existir apareixerà un *popup* indicant que insereixis grups vàlids. Fins que no s'insereixin grups vàlids no es permetrà continuar executant la funció (línia 52). En cas d'inserir grups vàlids s'intentarà executar la funció *cerca_horaris* del mòdul *filtro.py* que donarà com a resultat l'enviament del correu electrònic al correu electrònic inserit a la pantalla de presentació i a més a més cridarà la funció *popitup* de la que s'ha parlat abans (línia 30). En cas de no haver inserit un correu electrònic vàlid (inexistent i que genera l'error *SMTPRecipientsRefused*) s'obrirà un *popup* que a través d'un botó permetrà cridar a la funció *change1* (línia 22) i canviar a la pantalla inicial per tal d'inserir-ne un de vàlid.

Com s'ha pogut veure en aquest punt de la memòria la creació d'una pantalla que depèn de les seleccions d'una altre no és un cas trivial, i és la part que ha portat més temps en el desenvolupament de l'aplicació, per tema d'enllaçar diferents mòduls, diferents tipus de *widget* o diferents tipus d'error.

7. Planificació ETSEIB Calendar

7.1. Planificació temporal

En aquest punt es comentarà la planificació temporal que ha donat com a resultat l'aplicació ETSEIB Calendar. Cal tenir en compte que algunes tasques es van realitzar amb anterioritat a la matriculació del TFG per tal de veure si era possible la creació d'un gestor de calendaris. La totalitat de les tasques es poden veure al següent diagrama de Gantt:

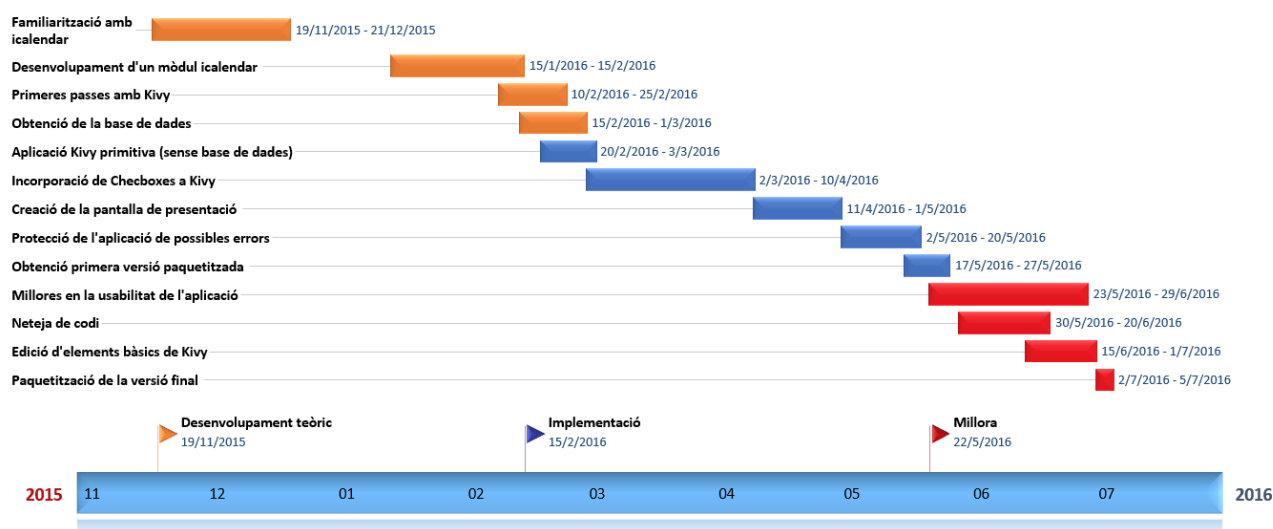


Figura 7.1: Diagrama Gantt d'ETSEIB Calendar

Resum de les tasques:

- Familiarització amb *iCalendar*. Per tal de veure les possibilitats de realització del projecte es va decidir estudiar la implementació d'arxius de naturalesa *iCalendar*. A través de l'estudi de la documentació es va veure que era possible llançar un projecte que treballés en aquest format per la creació d'un calendari de pràctiques.
- Desenvolupament d'un mòdul *iCalendar*. Un cop decidit que s'utilitzaria *iCalendar* es va cercar la possibilitat de treballar amb aquest arxius a través del mòdul *iCalendar* de *Python*. Aquest primer mòdul que es va desenvolupar era molt primitiu, i a partir d'un fitxer ".txt" amb dades com data, hora inici, durada i títol crearia al directori local un fitxer *iCalendar* que es podria importar a les plataformes de calendari (com poden ser Outlook o Google Calendar). Aquesta implementació de mòdul es la versió primitiva del que acabaria essent el mòdul "*arxicsmulti.py*".

- Primeres passes amb Kivy. Després de prendre la decisió de realitzar el projecte a través de la plataforma Kivy, va ser imprescindible llegir la documentació i veure tutorials per tal de veure quin abast real tenia plataforma de desenvolupament. Tenint en compte que l'aprenentatge es va iniciar des de 0, aquesta va ser una fase crítica del projecte.
- Obtenció de la base de dades. Es va demanar al suport informàtic de l'escola que es crees una url que permetés accedir als calendaris actualitzats de pràctiques. A través d'aquesta adreça web s'evitarien problemes d'actualització de dades i a més a més es va automatitzar el procés d'obtenció d'horaris de pràctiques. Com a resultat d'aquesta base de dades va sorgir el mòdul "*filtro.py*".
- Aplicació Kivy primitiva (sense base de dades). A partir d'un fitxer creat manualment amb horaris de pràctiques i una pantalla on s'inseririen els grups desitjats es va crear la versió primitiva d'ETSEIB Calendar. Es va decidir realitzar el prototip d'aplicació únicament amb 5 assignatures.
- Incorporació de Checkboxes a Kivy. Incorporant *checkboxes* a l'aplicatiu i la base de dades de pràctiques es va iniciar una evolució que donaria com a resultat la versió final de l'aplicació. Es va incorporar una pantalla de selecció d'assignatures on l'estudiant podria seleccionar entre les 13 possibilitats d'assignatura amb pràctiques de laboratori i com a pantalla final s'obtindria una de selecció de grups creada dinàmicament segons les seleccions de *checkboxes*. Aquesta pantalla final inicialment seguia una proporció de l'alçada de la pantalla entre totes les files i visualment no era agradable amb poques assignatures seleccionades.
- Creació de la pantalla de presentació. Es va decidir prescindir de la creació de l'arxiu contenidor de les pràctiques ("*.ics*") al directori de treball per temes de comoditat a la hora de trobar-ho i incorporar-ho a la plataforma calendari. Com alternativa es va proposar una pantalla inicial on s'explicaria el funcionament de la plataforma i a més a més existirien una sèrie d'inputs com ara el correu electrònic en el que es voldria rebre l'arxiu de pràctiques i a més a més l'anterioritat amb la que es volia rebre l'avís de pràctica de laboratori.
- Protecció de l'aplicació de possibles errors. Un cop creada l'estructura final de l'aplicació (Pantalla inicial - Selecció assignatures – Selecció de grups) es va analitzar el seguit d'errors que podrien ocasionar problemes a la hora d'executar l'aplicatiu i es va protegir amb la integració de comprovacions abans d'avançar a la següent pantalla dins del codi Python.
- Obtenció de la primera versió paquetitzada. Obtenció de la versió 0.1 d'ETSEIB

Calendar. Després de llegir la documentació de *Buildozer* fer diverses proves, es va paquetitzar l'aplicació i es va obtenir el primer arxiu ".apk" vàlid per el sistema operatiu Android.

- Millores en la usabilitat de l'aplicació. Es van integrar millores a l'aplicació com ara que els teclats de selecció de grups d'assignatura fossin numèrics o el teclat d'inserció del correu electrònic fos específic per correu electrònic (facilitant "@" i ".com"). A més a més es va solucionar la proporció d'alguns *labels* i botons, com ara els botons de la pantalla de selecció de grups (abans eren totalment proporcionals totes les files, es va aconseguir que una fila tingués una alçada fixa).
- Neteja de codi. Neteja i optimització del codi, ja que algunes variables o línies eren prescindibles.
- Edició d'elements bàsics de Kivy. Degut a que els acabats amb Kivy no eren del tot vistosos, es va decidir editar directament de la llibreria de Kivy alguns elements bàsics, com ara les *checkboxes*. Editant les dos imatges (seleccionat i no seleccionat) es va aconseguir millorar l'aspecte de l'aplicació i a més a més es va incorporar el botó d'informació a la pantalla inicial, que en comptes de ser un rectangle seria una imatge d'un interrogant.
- Paquetització de la versió final. Després d'incorporar totes les millores d'usabilitat i visuals a l'aplicació, es va decidir paquetitzar la versió final d'ETSEIB Calendar.

7.2. Planificació econòmica

En aquest punt es farà una aproximació dels costos associats al projecte, tenint en compte els elements que haurien de tenir una remuneració econòmica:

- S'aproxima el salari d'un enginyer junior en 15€/h. Les hores treballades per l'elaboració del software i la memòria corresponen amb 12 crèdits associats al Treball de Fi de Grau: si es considera que un crèdit correspon a 25 hores de treball, donaria com a resultat de 300 hores.
- S'estima en unes 50 hores el treball de supervisió realitzat per part de l'enginyer sènior. Aquestes hores tindrien un cost de 40€/h.
- El hardware utilitzat per la realització i desenvolupament de l'aplicació és un portàtil d'un cost de 800 euros. Es considerarà que el cicle de vida del portàtil és de 5 anys.

$$cost\ hardware = \frac{hores\ treballades}{5\ anys \cdot \frac{8760\ hores}{1\ any}} \cdot cost\ ordinador$$

Equació 7.1: Cost del hardware

- Els costos associats al transport i al material acadèmic que s'ha necessitat per la realització del projecte es de 30€.

Cal destacar que en aquesta llista de costos no s'han considerat els costos associats al software ja que Python i Kivy són plataformes gratuïtes. A més a més les hores de consum energètic que s'han utilitzat no s'han comptabilitzat.

Partida	Cost	Quantitat	Cost total (€)
Treball estudiant	15€/h	300	4500
Treball tutor	40€/h	50	2000
Hardware	800€	$\frac{1}{146}$	5,48
Transport	30€	N/A	30
TOTAL			6535,48

Taula 7.1: Resum de costos associats al projecte

8. Paquetització de l'aplicació

El fet d'utilitzar Kivy facilita tasques que d'altre manera serien més complicades de realitzar, com ara la creació de *widgets* que ja estan integrats dins la plataforma Kivy o la associació de determinats esdeveniments a altres funcions. Com s'ha vist al transcurs de la memòria a través del Kivy *language* es poden aconseguir en poques línies de codi grans resultats. Un altre punt a favor de la plataforma Kivy és la facilitat de la paquetització de l'aplicació en diferents plataformes. A la pàgina web de la Kivy està detallat tot el procés que caldria fer per importar-ho a altres plataformes, però s'ha estimat que per l'abast d'aquest projecte únicament es desenvoluparia en plataforma Android.

Per tal d'iniciar la paquetització de l'aplicació Kivy en extensió apk (extensió dels arxius instal·ladors d'aplicacions d'Android) cal seguir una sèrie de passos, que es comentaran detalladament en aquest punt:

1. Tenir instal·lat Buildozer a la màquina, ja que serà el programa que ajudarà a paquetitzar el conjunt dels arxius. En cas de no tenir-ho disponible caldrà aplicar les següents línies a la consola de Python :

```
git clone https://github.com/kivy/buildozer.git
cd buildozer
sudo python2.7 setup.py install
```

1. En un mateix directori cal tenir tots els arxius de l'aplicació, que en el cas d'ETSEIB Calendar serien els arxius d'extensió “.py” que contindrien totes les línies que permetrien realitzar tot el procés de l'aplicació. A més a més han d'estar al mateix directori totes les imatges utilitzades als diferents arxius, el logotip que es vol donar a l'aplicació i l'anomenat com *Presplash*, una imatge que sortirà quan es carregui l'aplicació.
2. Un cop situat tot al mateix directori, cal anomenar al arxiu principal de l'aplicació amb el nom *main.py*, ja que l'arxiu instal·lador es construirà a partir del bloc principal.
3. Ja a la consola de Python, dins del directori on estan tots els arxius caldrà executar la següent línia de codi:

```
buildozer init
```

4. Es crearà un arxiu automàticament al directori anomenat *buildozer.spec* que caldrà editar per personalitzar una sèrie d'aspectes de l'aplicació. Els canvis realitzats sobre la versió que es crea automàticament són:

- `title = ETSEIB Calendar`. Títol de l'aplicació que es veurà al

smartphone/tablet.

- `package.name = etseibcalendar`. Nom del paquet instal·lador de l'aplicació.
- `version = 0.8`. Versió de l'aplicació.
- `requirements = kivy, sqlite3, requests, simplejson, icalendar, datetime, pytz, HTMLParser, email, openssl`. Paquets que caldrà importar a l'arxiu instal·lador ja que no són predefinits a Python i en cas contrari no permetrien executar l'aplicació.
- `presplash.filename = loading.jpg`. Imatge que sortirà a la càrrega de l'aplicació.
- `icon.filename = icon.png`. Icona que tindrà l'aplicació al dispositiu Android.
- `orientation = landscape`. Orientació de l'aplicació al dispositiu (horitzontal).

5. Un cop editat l'arxiu *buildozer.spec* es pot procedir amb la paquetització, amb el següent codi:

```
buildozer android debug deploy
```

6. Es crearà una carpeta al directori anomenada bin, on estarà l'arxiu instal·lador que permetrà executar ETSEIB Calendar a qualsevol dispositiu amb una versió Android major a la 2.2.
7. Copiant l'arxiu instal·lador a un directori del dispositiu es podrà instal·lar fàcilment l'aplicació d'ETSEIB.

9. ETSEIB Calendar

Durant el transcurs de la memòria s'han tractat diferents aspectes tècnics de l'aplicació: com hauria de ser, que hauria de fer, alguns aspectes a tenir en compte... En aquest punt de la memòria es veurà per fi plasmada tota aquesta teoria en forma d'imatges i exemples.

Cal tenir en compte que el que es veurà en aquest punt és el resultat final de moltes iteracions, en les que s'han anat incorporant millores de codi i visuals que han donat forma al que es coneixerà com ETSEIB Calendar. La primera versió funcional està també detallada visualment a l'annex G, però hi ha hagut una gran evolució visual e incorporació de millores. A més a més de detallar les diferents pantalles es tractarà l'arbre d'errors, humans o tècnics, que es poden dur a terme. En el següents enllaços es pot trobar la primera versió i la versió definitiva de l'aplicació:

- https://www.dropbox.com/s/15j657yu6eni67h/ETSEIB_Calendar_V0.apk?dl=0
(primera versió funcional d'ETSEIB Calendar)
- https://www.dropbox.com/s/6ob00qlh9rbbnor/ETSEIB_Calendar_VFINAL.apk?dl=0
(versió definitiva d'ETSEIB Calendar)

9.1. Captures aplicació

Cal tenir en compte que al iniciar l'aplicació a una tablet o smartphone tardarà uns segons posar-se en marxa (es carreguen diferents mòduls de Kivy/Python). Durant aquest temps es mostra l'anomenada *Presplash Image* que ha sigut editada per la següent:

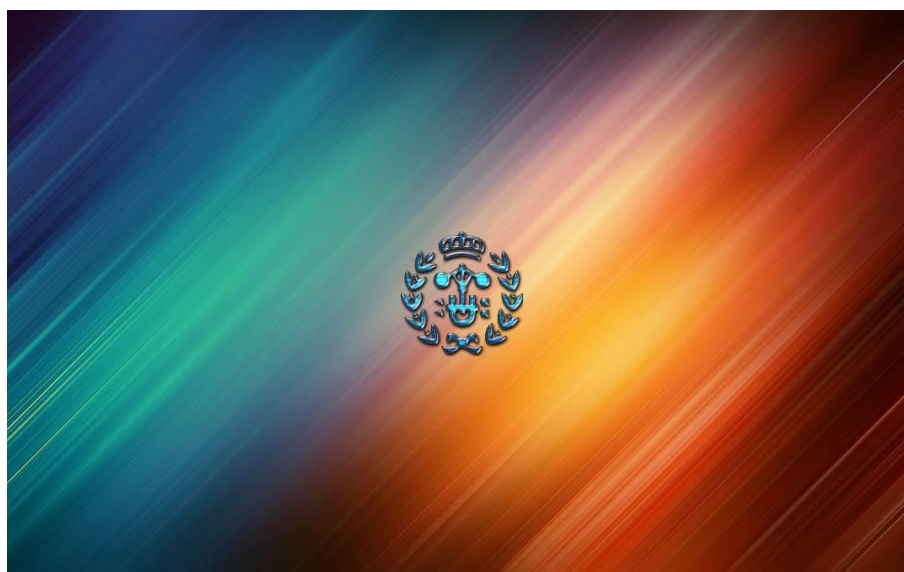


Figura 9.1: Pantalla de càrrega d'ETSEIB Calendar

Aquesta imatge s'adaptarà a la dimensió de la pantalla i potser no serà mostrada en la seva totalitat, però sí l'element central: el logotip de l'escola. Un cop carregada l'aplicació al nostre dispositiu es mostrarà la pantalla principal, un dels elements que més canvis ha tingut durant el procés de desenvolupament.

Aquesta pantalla principal és el resultat de molts iteracions, a continuació veurem un seguit de les versions més destacades, començant des de la simplicitat de la primera:

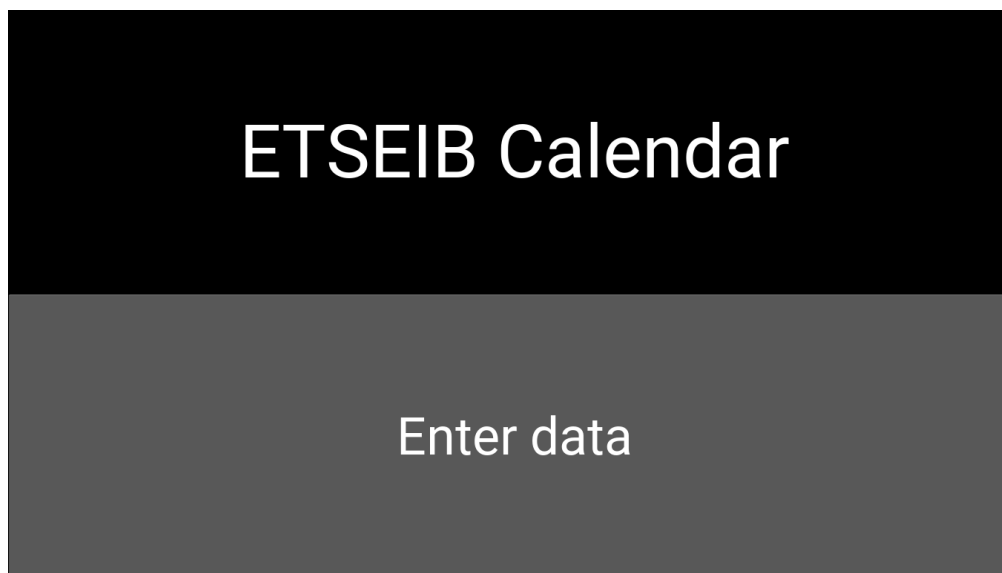


Figura 9.2: Primera versió pantalla inicial d'ETSEIB Calendar

Aquesta va ser la primera versió, la qual no tenia encara incorporada la possibilitat d'inserir el correu electrònic i l'antelació de l'avís. Aquesta versió va donar pas a la següent:

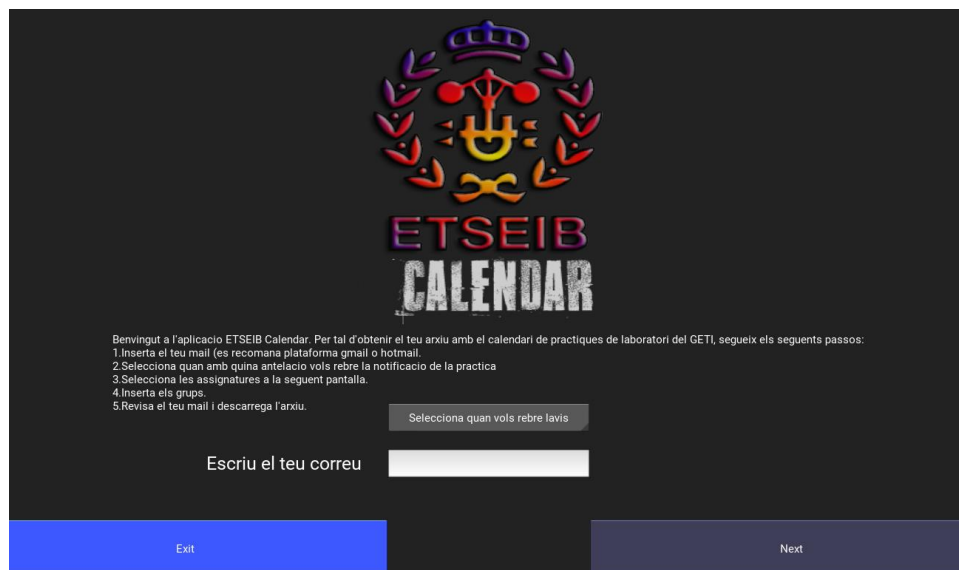


Figura 9.3: Segona versió de la pantalla inicial d'ETSEIB Calendar

Incloent l'*spinner* en el qual es podria seleccionar l'antelació de l'avis i el *textinput* on inserir el correu electrònic es va donar pas al principi de la versió final, però encara quedaven aspectes a millorar, com ara l'aspecte visual:

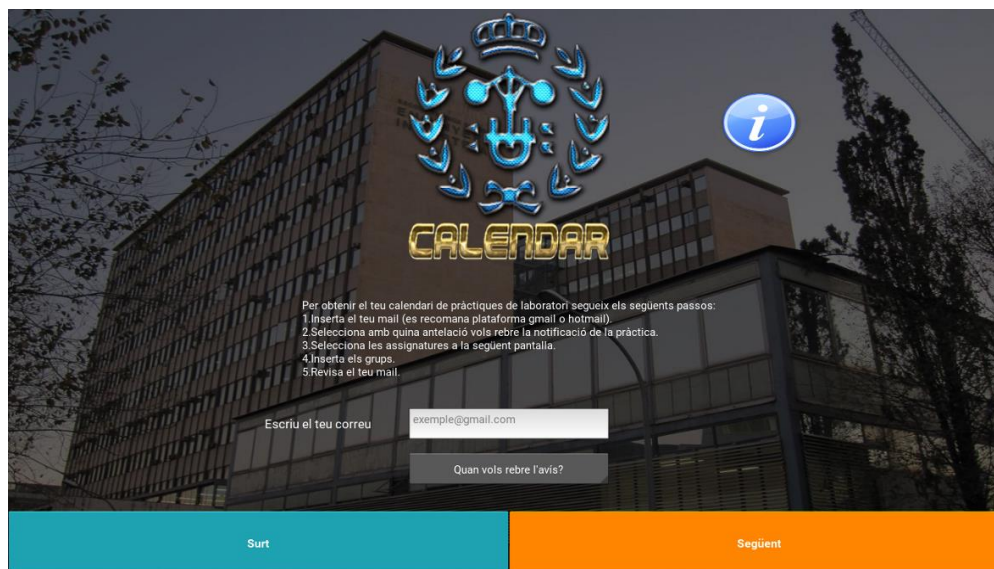


Figura 9.4: Tercera versió pantalla inicial d'ETSEIB Calendar

Es va integrar un fons a la pantalla principal on es pot veure l'escola i a més a més es va canviar el logotip i es va integrar el botó informació. A més a més el *textinput* permetia donar un exemple de correu i els quadres de text s'adaptaven a la majoria de pantalles (no a totes les pantalles). Això va comportar la creació d'una següent versió, seguint la mateixa estètica i organització, i que acabaria essent la versió definitiva de la pantalla principal:

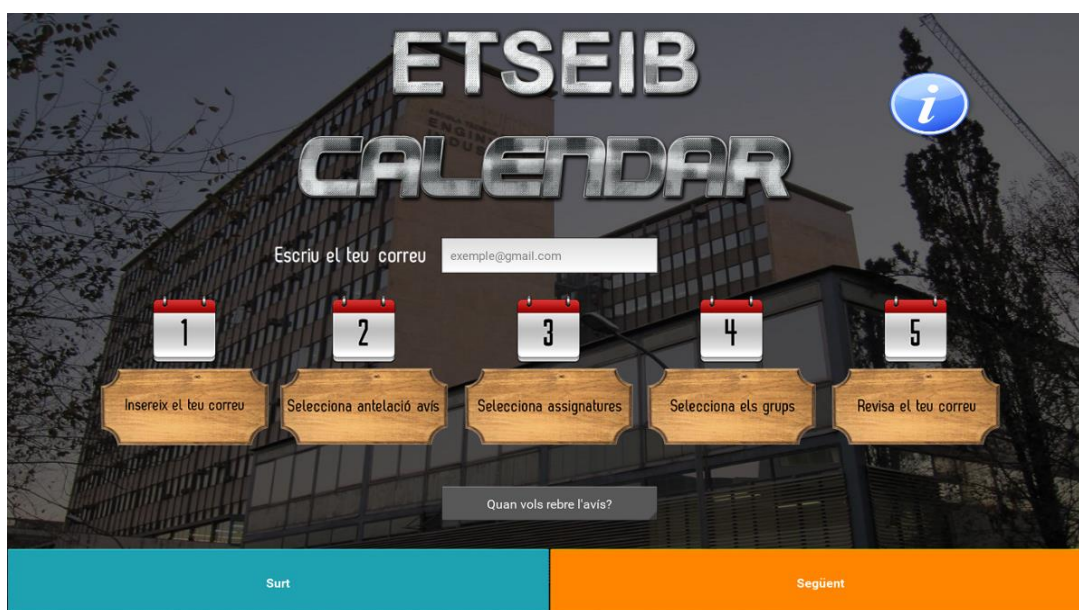


Figura 9.5: Versió final de pantalla inicial

Inserint imatges en comptes de quadres de text s'evitaven problemes d'adaptació a amplades de pantalles molt petites. A més a més en cas de que el teclat amagués el quadre de text on s'insereix el correu electrònic, la pantalla es mouria al dispositiu mòbil per facilitar la visualització del que s'està escrivint. Una altre millorar que es va incorporar respecte les versions anteriors va esser el fet de tenir un teclat específic per correu electrònic, que permetria tenir elements com “@” o “.com” assolibles de forma directe al nostre dispositiu mòbil. Si es selecciona el botó de informació (botó de Kivy editat per tenir aquest aspecte) es mostrarà el següent *popup*:

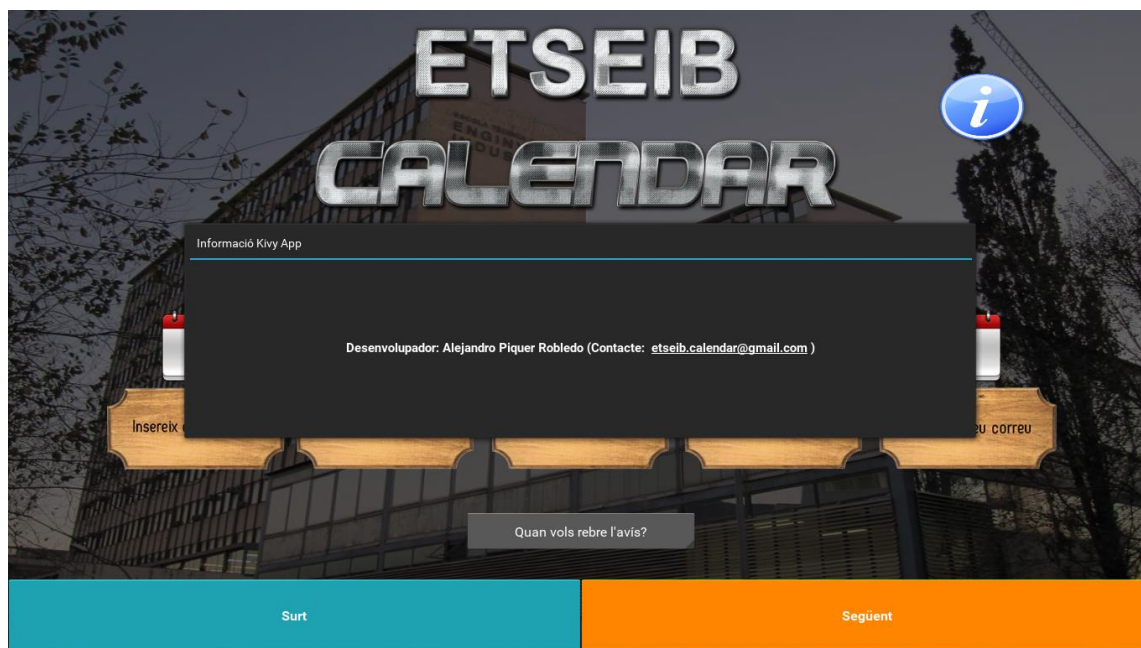


Figura 9.6: Captura del Popup informatiu d'ETSEIB Calendar

Inserint els camps d'antelació i de correu electrònic i seleccionat el botó “Següent” s'avançarà a la següent pantalla, en cas de no fer-ho s'executarà l'error 1. En cas de no disposar de connexió a Internet s'executarà l'error 2. Si s'han seleccionat correctament els camps i es disposa de connexió a la xarxa s'avançarà a la següent pantalla, on es podran seleccionar entre les 7 primeres assignatures que disposen de pràctiques de laboratori del GETI:

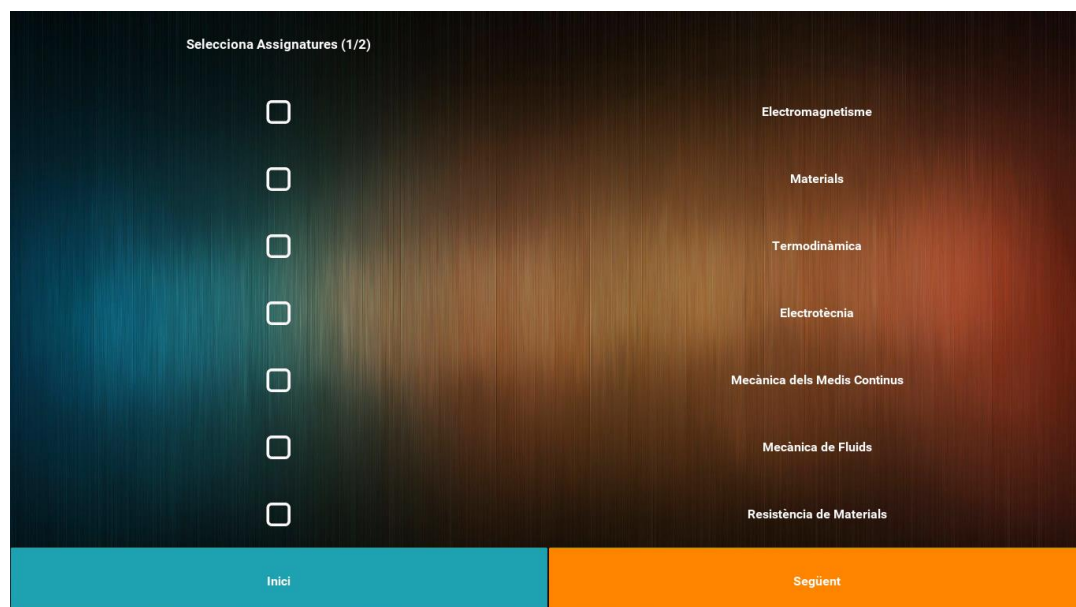


Figura 9.7: Captura de de la primera pantalla de selecció d'assignatures d'ETSEIB Calendar

Com es pot veure les *checkboxes* utilitzades no són les pre-definides de Kivy, ja que es va considerar que per l'aspecte visual de l'aplicació eren massa petites. Si es desitja anar a les següents 6 assignatures es pitjarà el botó següent i apareixerà la següent pantalla:

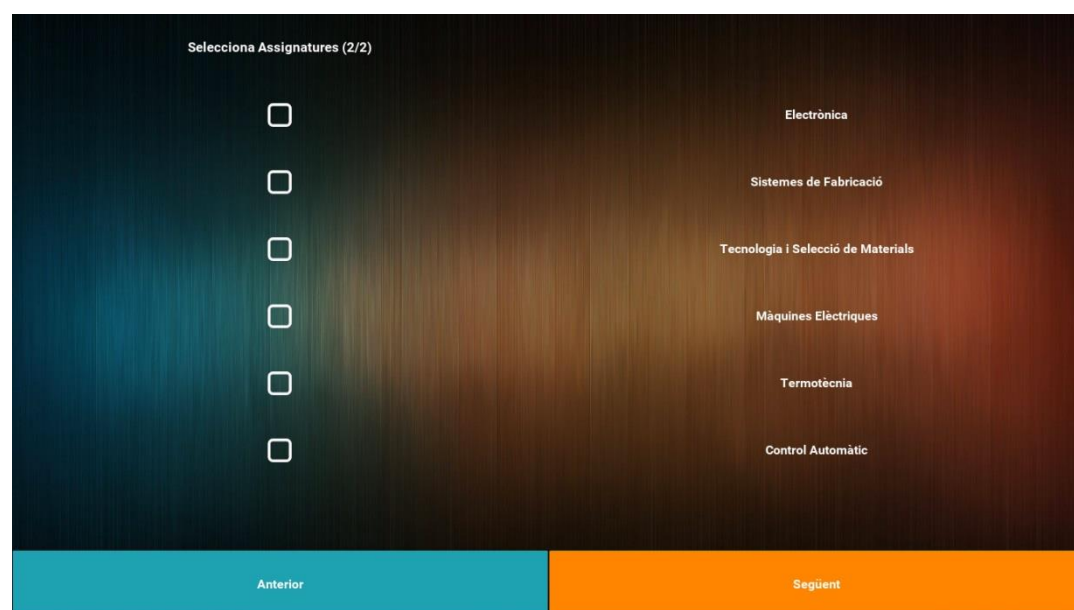


Figura 9.8: Captura de de la segona pantalla de selecció d'assignatures d'ETSEIB Calendar

En cas de no seleccionar cap assignatura s'executarà l'error 3. És obvi que si es vol un calendari de pràctiques cal seleccionar alguna assignatura, per tant aquest error serà poc

comú. Si es fan les següents seleccions:

- Materials
- Electrotècnia
- Mecànica dels medis continus
- Sistemes de fabricació
- Termotècnia

Es crearà una pantalla d'inserció d'assignatures com la següent (en cas de no disposar de connexió a Internet, ja que es pot haver perdut durant la selecció s'accionarà l'error 4):

Assignatura	Escriu el grup
240033-Materials	11, 12, 21, 22, 31, 32, 41, 42
240053-Electrotècnia	11, 12, 13, 14, 21, 22, 23, 24, 31, 32, 33, 34, 41, 42, 43, 44
240054-Mecànica dels Medis Continus	11, 12, 13, 14, 21, 22, 23, 24, 31, 32, 33, 41, 42, 43
240073-Sistemes de Fabricació	11, 12, 13, 14, 21, 22, 23, 24, 31, 32, 33, 34, 41, 42, 43, 44
240171-Termotècnia	11, 12, 13, 14, 21, 22, 23, 24, 31, 32, 33, 34, 41, 42, 43, 44
Anterior	Executa

Figura 9.9: Captura de de la pantalla de selecció de grups d'ETSEIB Calendar

Es pot veure com es poden pre-visualitzar els grups possibles per cada assignatura seleccionada, i a més a més el teclat serà del tipus numèric, a diferència del de la primera pantalla. Si es seleccionés al dispositiu mòbil l'última assignatura la pantalla es desplaçaria per poder visualitzar que s'està escrivint (aquesta és una de les últimes que es van incorporar). Cal tenir en compte que les primeres versions de l'aplicació no donaven una proporció diferent a la última fila de la quadrícula i que el fet de poder seguir les mateixes proporcions que la resta de pantalles va representar un gran avenç visual. Si es seleccionen grups no vàlids (inexistents o deixar el *textinput* buit) apareixeria l'error 5. Inserint tots els grups correctament i pitjant el botó "Executa" es durà a terme la creació del calendari, que enviarà per correu

electrònic l'arxiu contenidor dels horaris de pràctiques amb l'antelació seleccionada. Si s'hagués inserit un correu electrònic incorrecte (inexistent) s'executarà l'error 6, que permetrà anar directament a la pantalla inicial per modificar el correu inserit. Si el procés s'ha executat correctament apareixerà el següent missatge:

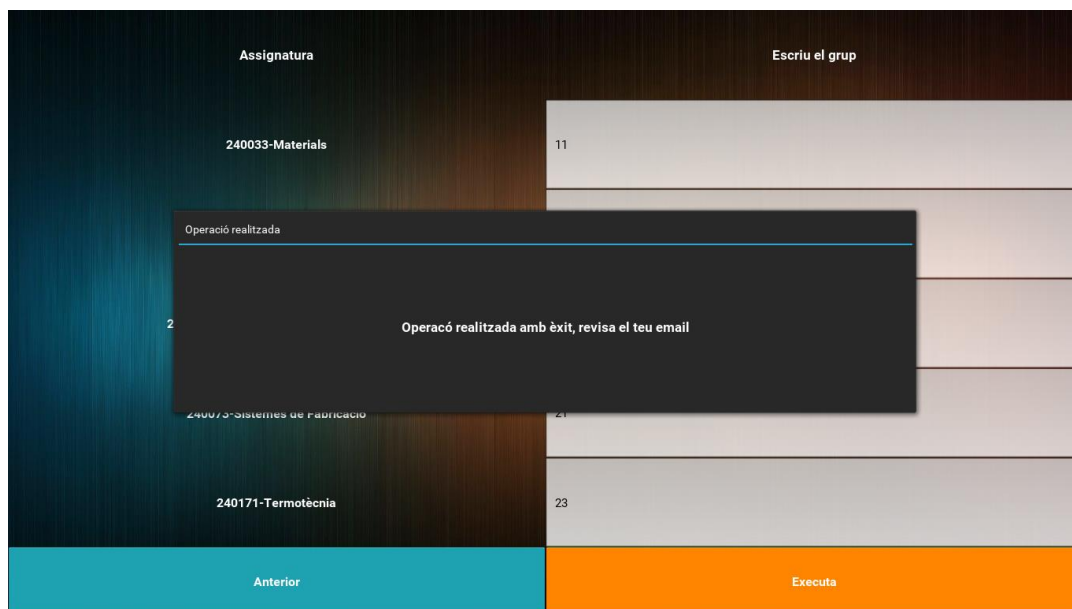


Figura 9.10: Captura del Popup d'operació realitzada d'ETSEIB Calendar

9.2. Arbre d'errors

Tal i com s'ha parlat al punt anterior, els possibles errors que es poden tenir durant l'execució de l'aplicació són:

1. No inserir el correu electrònic i/o no seleccionar l'antelació de l'avís de la pràctica.
2. No disposar de connexió a la xarxa i per tant a la base de dades de les pràctiques de laboratori.
3. No seleccionar cap assignatura del llistat d'assignatures que disposen de pràctiques de laboratori.
4. No tenir accessible la connexió a Internet en el moment d'intentar anar a la pàgina de selecció de grups.
5. No inserir grups vàlids als *textinputs* de la pantalla de selecció de grups.
6. No haver inserit un correu electrònic vàlid a la pàgina inicial.

Cal tenir en compte que qualsevol tipus d'error impossibilitarà avançar a la següent pantalla fins que no sigui solucionat, per tant es pot dir que per poder dur a terme tot el procés caldrà evitar les 6 possibilitats d'error. Als següents gràfics es pot veure de forma detallada la relació dels errors amb el funcionament de l'aplicació, segons la pantalla en la que estiguem:

- Pantalla Presentació:

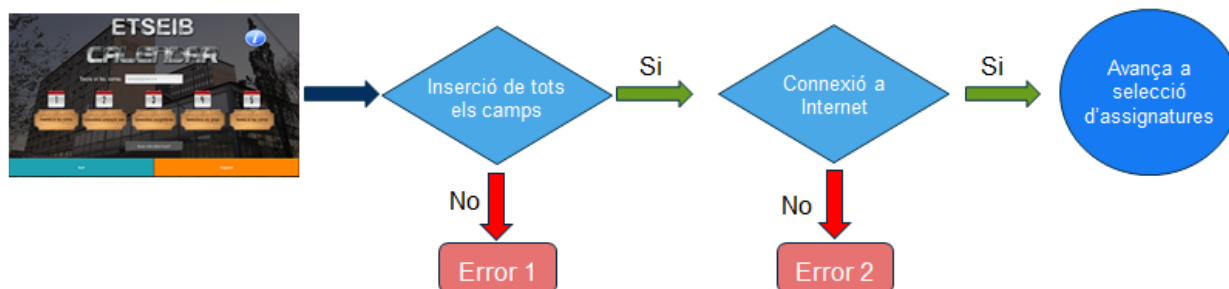


Figura 9.11: Esquema de funcionament de la primera pantalla

L'error 1 s'indica amb el següent avís de *popup*:

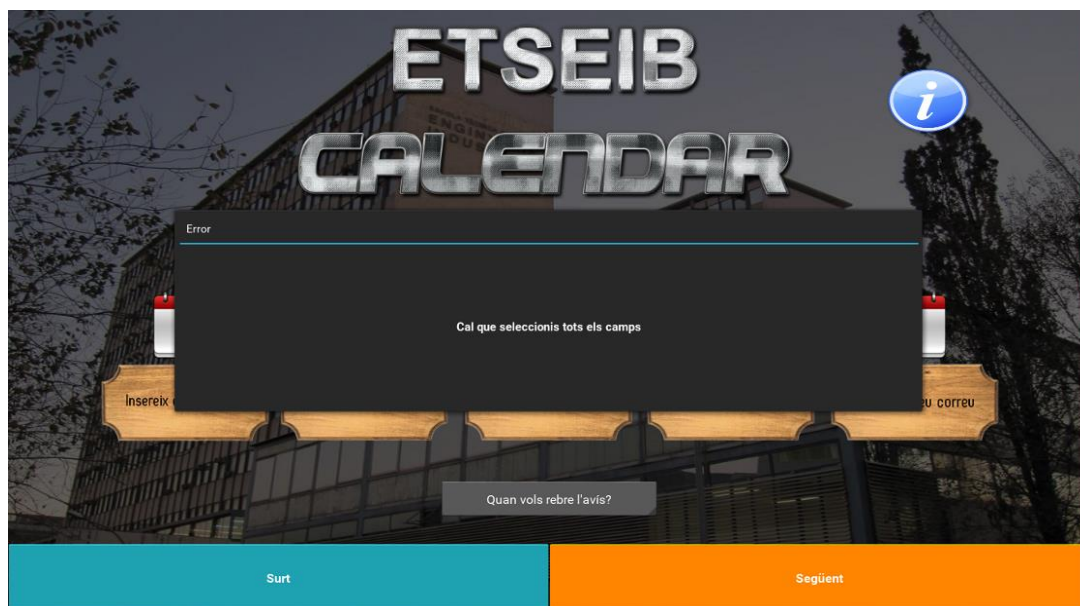


Figura 9.12: Popup corresponent a l'error 1

Per tal d'evitar aquest error caldrà seleccionar l'antelació de l'avís que es vol rebre de la pràctica de laboratori i el correu electrònic que es vol utilitzar per enviar l'arxiu contenidor de pràctiques.

Si s'han complimentat correctament els camps es comprovarà la connexió a Internet, que en cas d'ésser inexistent farà que aparegui el següent missatge:

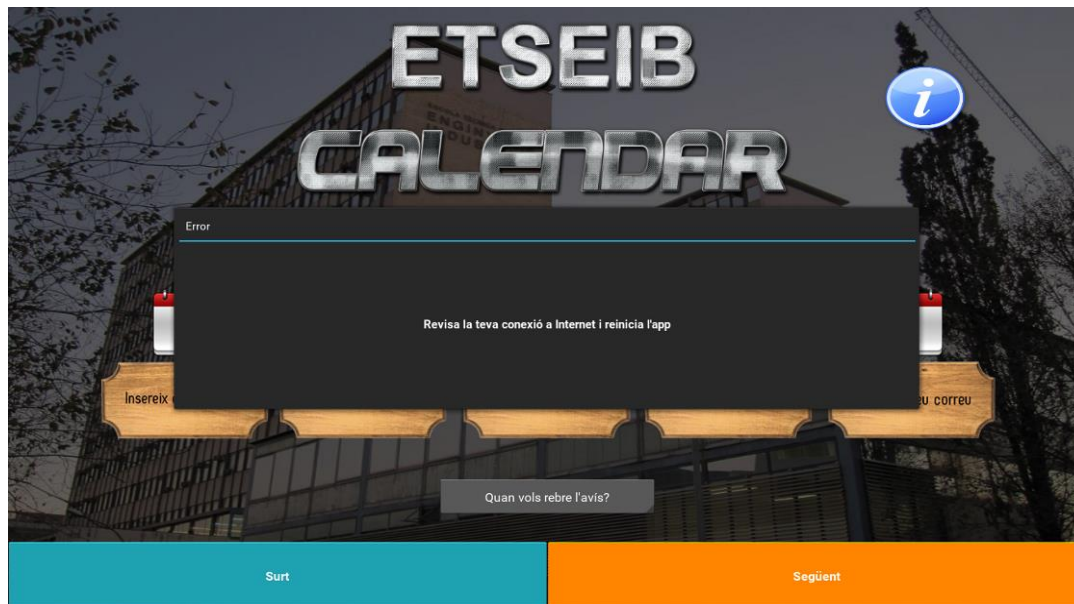


Figura 9.13: Popup corresponent a l'error 2

Si es vol evitar aquest error caldrà disposar de connexió a Internet, ja que l'aplicació utilitzarà la xarxa durant el procés d'obtenció dels horaris.

- Pantalla de selecció d'assignatures:

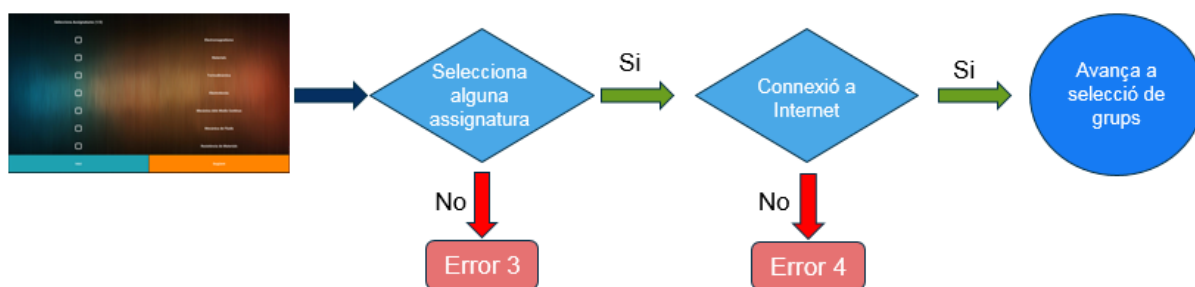


Figura 9.14: Esquema de funcionament de les pantalles de selecció d'assignatures

L'error 3 donarà com a resultat el següent *popup*:

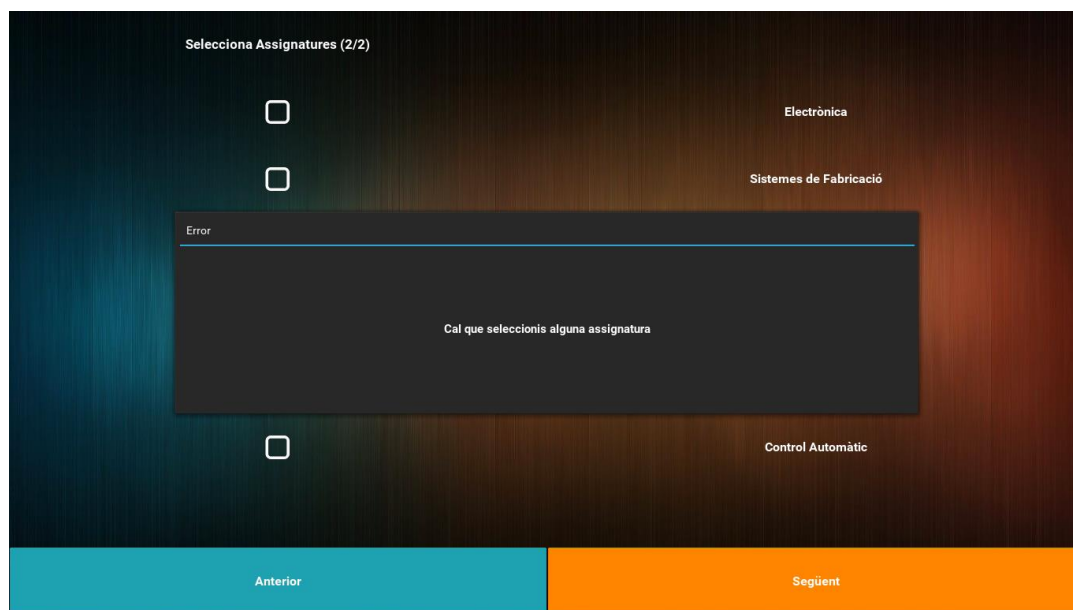


Figura 9.15: Popup corresponent a l'error 3

Seleccionant una assignatura com a mínim s'evitarà aquest error. A més a més es revisarà la connexió a Internet de nou i en cas de no existir s'accionarà l'error 4:

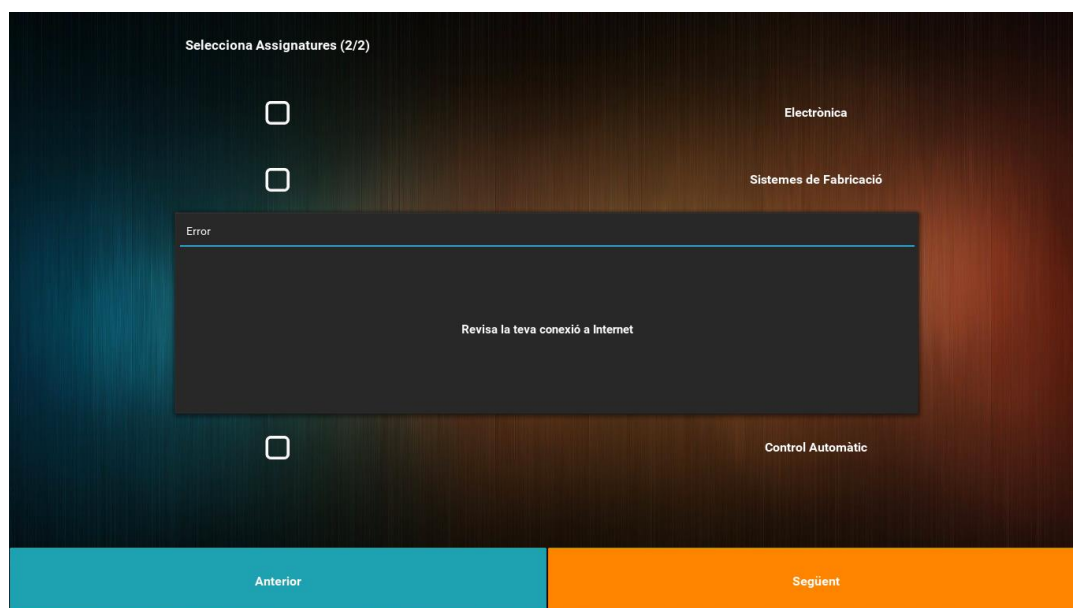


Figura 9.16: Popup corresponent a l'error 4

- Pantalla de selecció de grups:

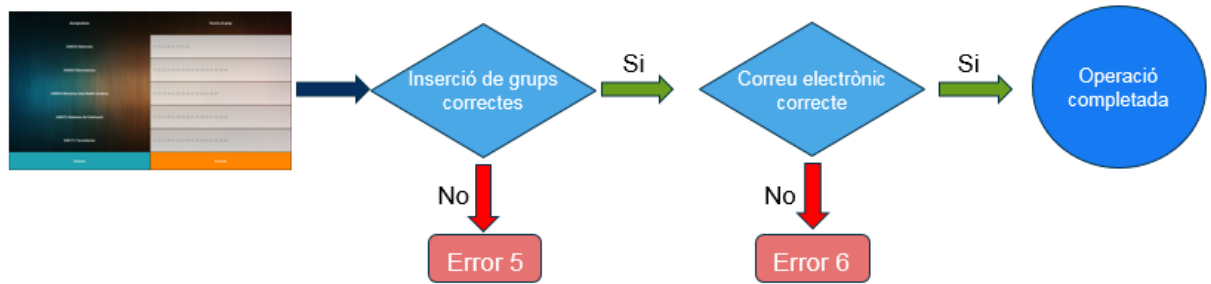


Figura 9.17: Esquema de funcionament de la pantalla de selecció de grups

En cas d'inserir grups no vàlids (inexistents o buits) s'accionarà l'error 5, llançant aquest *popup*:

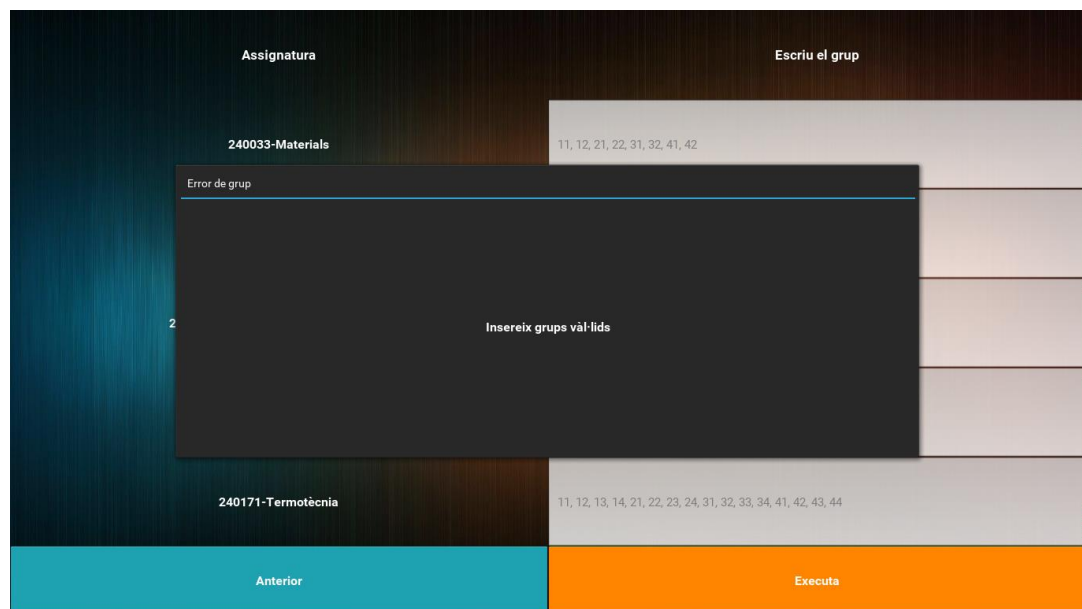


Figura 9.18: Popup corresponent a l'error 5

Seguint les indicacions dels *textinputs* es pot saber quines possibilitats de grups es poden seleccionar, facilitant així la tasca de l'usuari. Però cal tenir en compte la possibilitat de que s'hagi inserit a la primera pantalla un correu electrònic inexistent, que accionaria l'últim error, el 6:

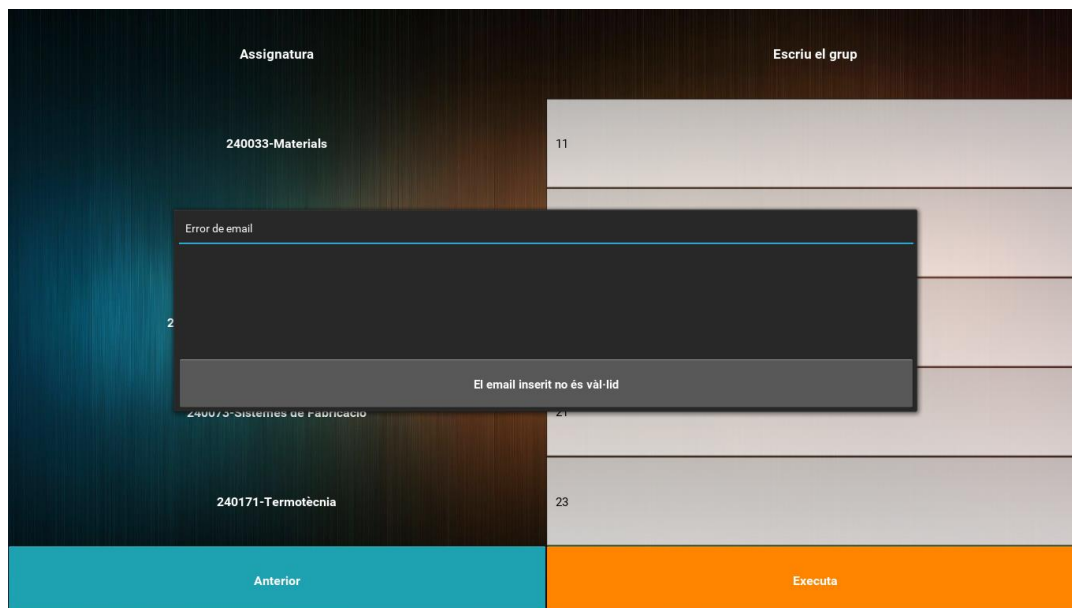


Figura 9.19: Popup corresponent a l'error 6

Aquest *popup* es diferencia amb els altres en que no és un quadre de text, sinó un botó que permetrà anar a la pantalla inicial e inserir un altre correu electrònic vàlid. Encara que visualment no sigui molt vistós, és necessari per l'aplicació.

9.3. Tramitació de les sol·licituds de calendari

Durant el transcurs de la memòria s'ha parlat de l'evolució de la tramitació de sol·licituds de calendari de pràctiques de laboratori, però no s'ha parlat explícitament del funcionament de la versió final d'ETSEIB Calendar. Inicialment es creava un arxiu *iCalendar* al directori de l'ordinador però amb el pas del temps es va decidir prescindir de la creació en local, ja que després a la hora d'importar-ho a les plataformes de calendari calia buscar el directori i només era possible treballant des d'un ordinador, no des d'un *smartphone*. Per tal d'automatitzar aquest procés es va decidir crear un correu electrònic que serviria per enviar els arxius contenidors i a més a més permetria realitzar un control estadístic de les sol·licituds de calendaris. El domini escollit va ser gmail i el correu és el següent:

etseib.calendar@gmail.com

A més a més aquesta direcció serviria per els usuaris que es vulguin adreçar amb qualsevol dubte o iniciativa de millora a la plataforma, atorgant així als usuaris el poder de realitzar un *feedback* sobre l'aplicació.

9.4. Exemple ETSEIB Calendar

Per tal de fer més entenedor el procés que caldria fer per obtenir en la seva totalitat un calendari de pràctiques de laboratori en aquest punt es realitzarà un exemple d'un calendari de pràctiques per les següents característiques:

- Correu electrònic: alex.piquer@hotmail.com
- Antelació de l'avís: 1 hora abans
- Assignatures i grups:
 - Materials (Grup: 11)
 - Electrotècnia (Grup: 12)
 - Mecànica dels medis continus (Grup: 13)
 - Sistemes de fabricació (Grup: 21)
 - Termotècnia (Grup: 23)

La simulació amb aquestes dades donaria com a resultat les següents captures de pantalla:

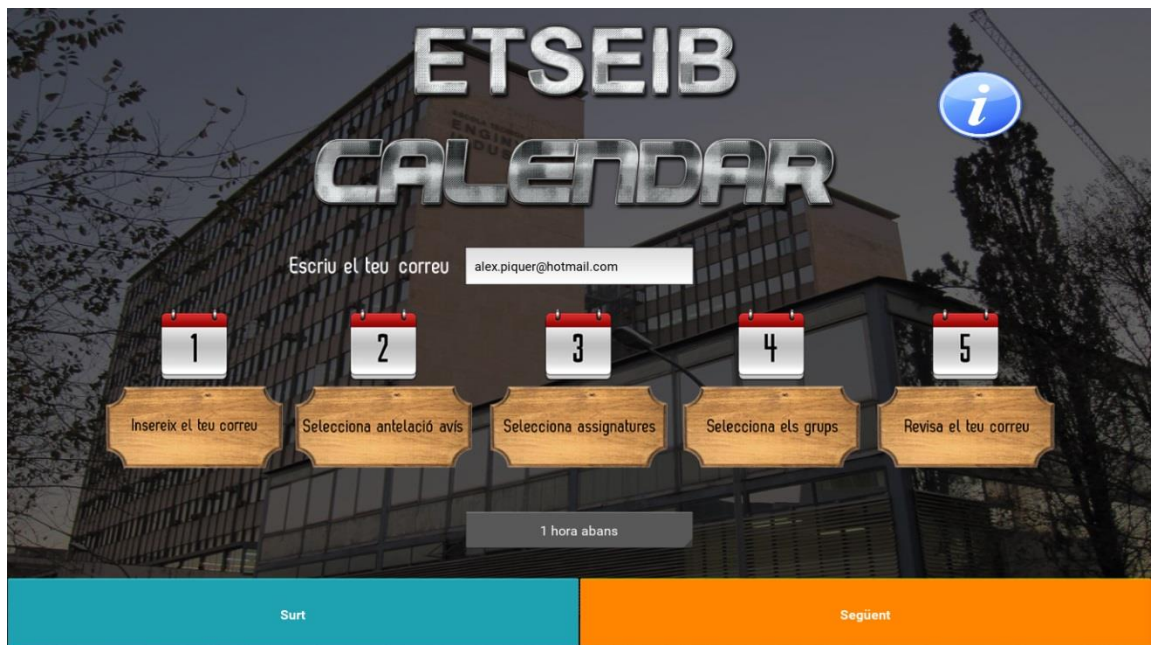


Figura 9.20: Primera pantalla amb dades inserides

A la pantalla principal s'inseriria el correu electrònic i l'antelació de l'avís respecte la pràctica.

Selecciona Assignatures (1/2)

<input type="checkbox"/>	Electromagnetisme
<input checked="" type="checkbox"/>	Materials
<input type="checkbox"/>	Termodinàmica
<input checked="" type="checkbox"/>	Electrotècnia
<input checked="" type="checkbox"/>	Mecànica dels Medis Continus
<input type="checkbox"/>	Mecànica de Fluids
<input type="checkbox"/>	Resistència de Materials

Inici Següent

Figura 9.21: Primer bloc d'assignatures amb seleccions

Es seleccionen les assignatures que apareixen a la primera pantalla de selecció.

Selecciona Assignatures (2/2)

<input type="checkbox"/>	Electrònica
<input checked="" type="checkbox"/>	Sistemes de Fabricació
<input type="checkbox"/>	Tecnologia i Selecció de Materials
<input type="checkbox"/>	Màquines Elèctriques
<input checked="" type="checkbox"/>	Termotècnia
<input type="checkbox"/>	Control Automàtic

Anterior Següent

Figura 9.22: Segon bloc d'assignatures amb seleccions

Un cop seleccionades la totalitat d'assignatures obtindrem la següent pantalla de grups, amb el grups els quals ens agradaria rebre les notificacions de pràctica:

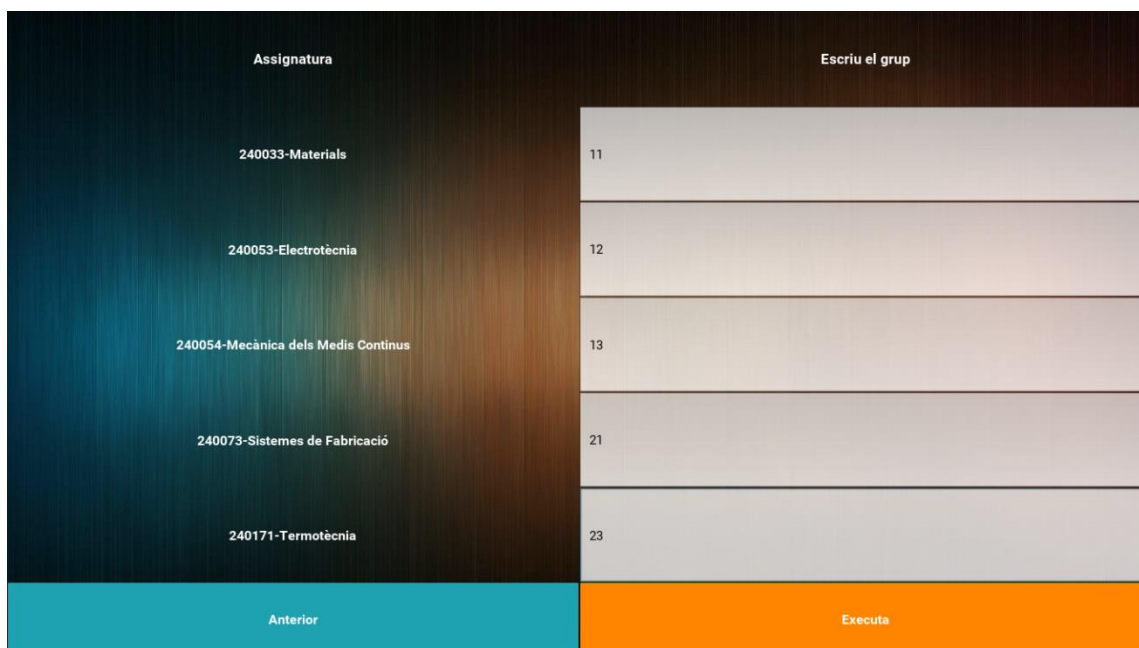


Figura 9.23: Pantalla de selecció de grups amb grups inserits

I obtindríem el següent missatge un cop el procés s'hagués realitzat correctament:

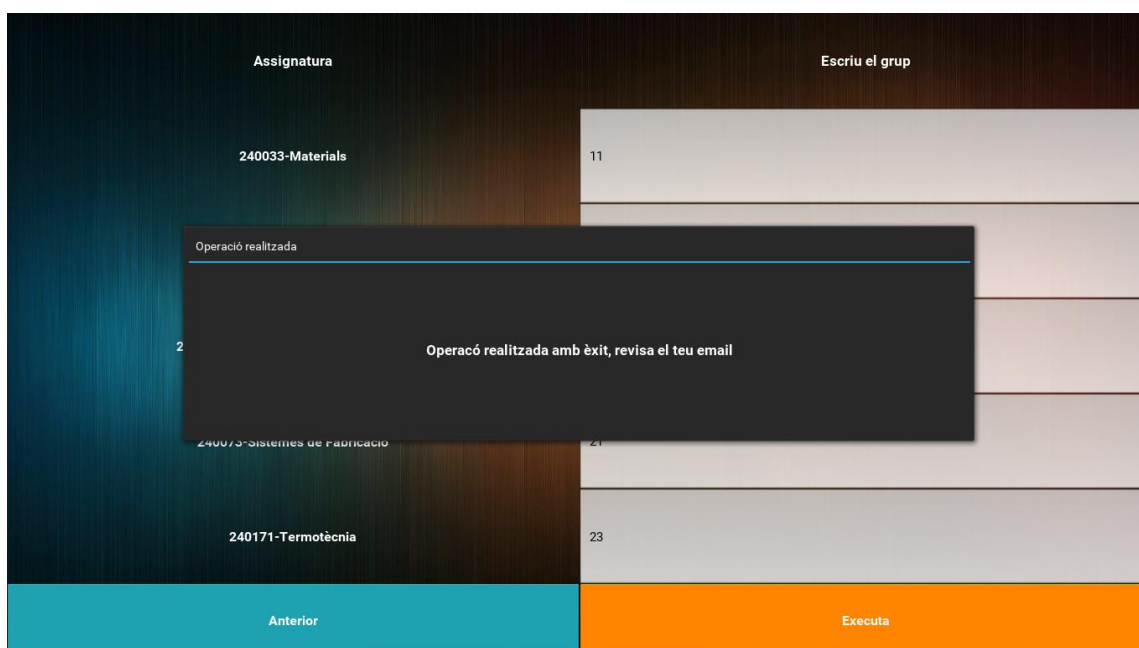
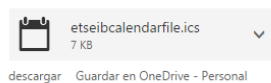


Figura 9.24: Missatge d'operació realitzada amb èxit

Revisant el correu electrònic trobaríem un email provinent del compte etseib.calendar@gmail.com. Aquest compte ha estat creat específicament per l'enviament

d'arxius *iCalendar*, i des d'ell s'envien tots els arxius contenidors de calendaris. El correu tindrà el següent format:



Instruccions per importar l'arxiu icalendar a Google Calendar:

- 1: Descarrega l'arxiu adjunt del email.
- 2: Obre Google calendar en un ordinador (només es pot importar calendaris des d'un ordinador).
- 3: Adalt a la dreta, fes clic a configuració.
- 4: Obre la pestanya de Calendaris.
- 5: Fes clic a Importar calendaris.
- 6: Fes clic a selecciona arxiu i escull l'arxiu que has descarregat anteriorment.
- 7: Escull a quin calendari desitges importar-ho.
- 8: Fes clic a Importar.

Instruccions per importar l'arxiu icalendar a Outlook:

- 1: Descarrega l'arxiu adjunt del email.
- 2: Obre Outlook en un ordinador (només es pot importar calendaris des d'un ordinador).
- 3: Adalt a la esquerra, fes clic a aplicacions.
- 4: Selecciona Calendari.
- 5: Fes clic a Agregar Calendaris.
- 6: Fes clic a "Desde un arxiu" i escull l'arxiu que has descarregat anteriorment.
- 7: Clica a Guardar.

Figura 9.25: Correu electrònic que es rebrà amb l'arxiu contenidor

Amb l'última actualització d'Outlook es permeten incorporar directament els esdeveniments al calendari, fent la següent tasca :

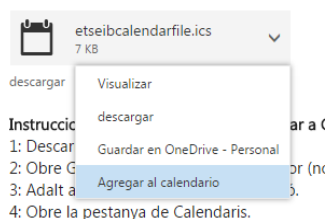


Figura 9.26: Possibilitats que dona Outlook al rebre el correu electrònic

La notificació que es rebrà serà com la següent:

Evento: lun 14/03/2016, de 12:00 a 17:00 | Etseib H-H 7.56 | [Ver detalles del evento](#)

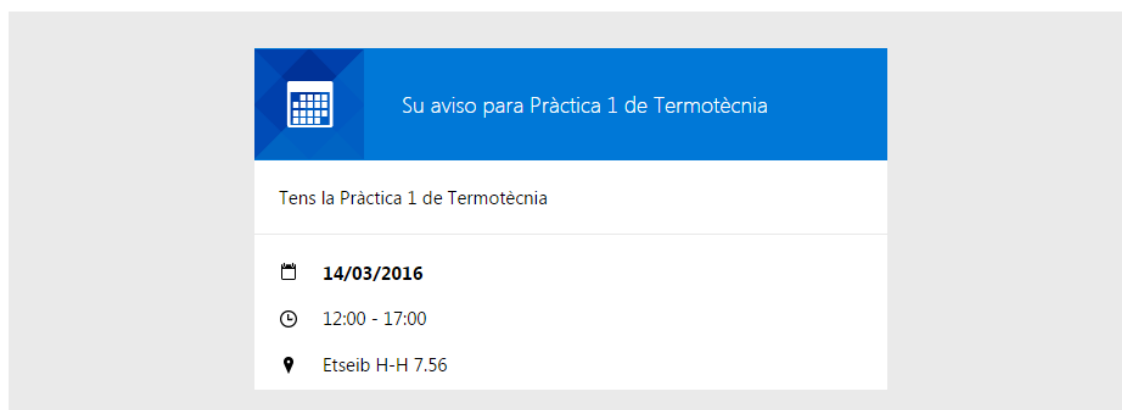


Figura 9.27: Avís que es rebrà amb l'antelació seleccionada

Com es pot veure s'informa de la pràctica, la hora, el dia i l'aula on s'haurà de realitzar.

Cal tenir en compte que es va començar a treballar amb les dades de les pràctiques del quadrimestre anterior, per tant, a la hora de realitzar qualsevol calendari, si s'utilitza la base de dades desactualitzada, totes les notificacions de pràctiques arribaran en bloc. S'intentarà gestionar amb el servei informàtic de les pràctiques un manteniment de la base de dades, per tal de no estar desactualitzat.

9.5. Control estadístic

Per tal de portar un control estadístic de la plataforma inicialment es va estimar realitzar-ho a través dels *Spreadsheet* de Google, els quals, a través de codi Python permetrien emmagatzemar en una base de dades al núvol totes les sol·licituds de calendaris tramitades, guardant la informació que sigui d'utilitat per un futur anàlisi estadístic (en quin període de l'any s'utilitza més l'aplicació, quines assignatures són més seleccionades...). Aquesta idea, però, s'escapava en quant a temps i coneixements en la escriptura de *spreadsheets* (un full d'Excel per exemple que emmagatzemem totes les dades) . Finalment es va decidir treballar amb una plataforma ja existent anomenada Gmail Meter, la qual permetria analitzar algunes dades d'interès en el període desitjat (trimestralment, mensualment, setmanalment...). Potser no aportaria tota la informació que es podria obtenir desenvolupant el codi manualment, però donaria una petita idea de l'ús d'ETSEIB Calendar.

A través del compte de correu electrònic encarregat de tramitar les sol·licituds de calendari (etseib.calendar@gmail.com) s'analitzarien les tramitacions realitzades durant el període estimat i es plasmarien a través d'alguns indicadors que serien enviats com un *report* a la mateixa compte de correu electrònic. A continuació es pot veure un exemple d'informe:

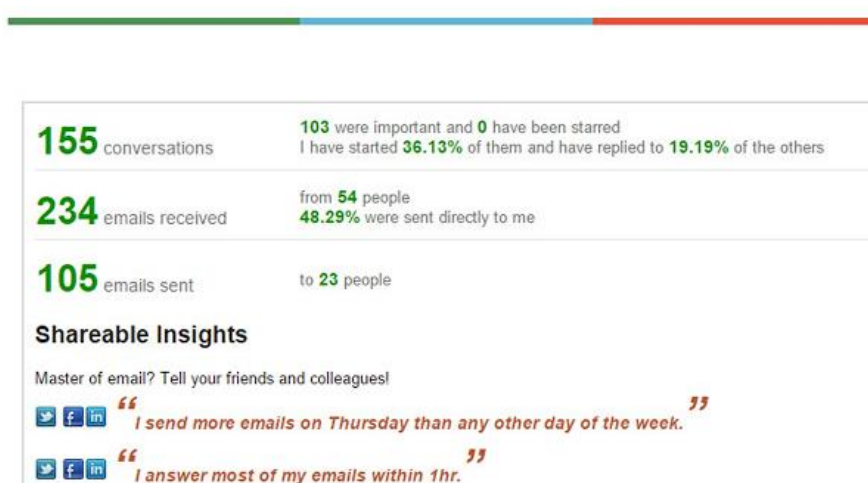


Figura 9.28: Exemple d'anàlisi estadístic que aporta Gmail Meter

Com es pot veure la plataforma aporta dades que poden ajudar a veure la magnitud que té la plataforma, com els correus enviats (cada correu és un calendari), a quantes persones... A més a més aporta resums gràfics:



Figura 9.29: Exemples de resums gràfics de Gmail Meter

La plataforma incorpora moltes altres possibilitats de gràfics i resums, que temporalment podran cobrir l'ús d'un algoritme que crees automàticament una base de dades al núvol.

10. Proposta de continuació del treball

El projecte ETSEIB Calendar va ésser desenvolupat amb l'objectiu d'esdevenir un prototip a una futura incorporació a la plataforma de calendaris i horaris de l'actual pàgina web de l'escola. En aquest punt s'estudiaran algunes possibilitats de continuació d'aquesta idea, com podrien ser:

1. Convertir l'aplicació a altres plataformes, com podrien ser Windows, Mac o IOS. Tot aquest projecte ha acabat sent desenvolupat en la plataforma d'Android, ja que actualment la majoria de *smartphones* treballen en aquesta plataforma. Però, afegint més possibilitats d'execució per ETSEIB Calendar, es podria arribar a un públic més extens i es guanyarien més usuaris. Per limitacions temporals no es va acabar de profunditzar en altres plataformes, per tant, queda oberta la possibilitat d'ampliar el camp de treball d'ETSEIB Calendar.
2. Incorporar la idea d'ETSEIB Calendar a la pàgina web de l'escola. Actualment dintre de l'apartat "Portal d'assignatures i horaris" de la web de l'escola trobem el simulador d'horaris. Una possible incorporació de l'algoritme de l'aplicació a la pàgina web suposaria guanyar en comoditat a la hora de simular els calendaris d'assignatures de cada estudiant. Amb una incorporació d'un botó de creació d'arxiu *iCalendar* es podria facilitar molt la tasca de molts estudiants i a més a més suposaria un gran avenç tecnològic.
3. Millorar les prestacions de Kivy desenvolupant l'aplicació a través d'altres llenguatges de programació, com ara *javascript* o codi *html*. Creant una pàgina web es podria aconseguir arribar a la majoria dels dispositius i a més a més solucionaria problemes de conversió a altres plataformes. Actualment es disposa de moltes possibilitats pel desenvolupament d'aplicacions i totes disposen de punts a favor i punts en contra.
4. Incorporar a ETSEIB Calendar els horaris de les assignatures i així ampliar el rang de treball de les pràctiques de laboratori a totes les assignatures del GETI. A través de la previsualització prèvia dels horaris (actualment la eina no ho permet) es podrien advertir possibles solapaments. Així els estudiants de GETI disposarien d'una eina amb la que simular horaris i obtenir arxius contenidors, atorgant una alternativa al portal d'assignatures i horaris dins de la plataforma Android. A més a més es podria programar que cada cop que s'executés una tramitació de calendari les dades utilitzades fossin emmagatzemades al núvol i així es podria portar un control estadístic dels usuaris per extreure conclusions certes sobre la utilitat de la plataforma. Això permetria rebre un *feedback* de la tramitació de calendaris.

Conclusions

Com s'ha pogut comprovar en el transcurs de la memòria ETSEIB Calendar ha passat de ser una idea a una realitat. Aquest projecte tal i com es va comentar a la introducció no té una altre fi que facilitar les tasques de gestió de pràctiques de laboratori, ja que en molts casos els estudiants perdem temps en apuntar les pràctiques de laboratori i amb aquesta plataforma es podria realitzar automàticament. Des d'un inici es va pensar en una idea que utilitzes com a eines de desenvolupament les adquirides durant la formació universitària. A través dels coneixements de Python i amb la introducció de la plataforma Kivy, per tal de facilitar la creació de l'aplicació, s'ha obtingut un aplicatiu Android amb el qual tots els estudiants podran rebre un calendari de pràctiques importable als seus correus electrònics.

El camí ha estat llarg fins arribar fins aquí, ja que el fet d'afrontar una plataforma totalment nova com era Kivy va suposar un estudi de la documentació que va comportar una gran inversió de temps. El resultat, però, parlen per si sols, a dia d'avui ETSEIB Calendar és totalment funcional i a partir d'una sèrie d'inputs senzills com ara l'antelació de l'avís, el correu electrònic, les assignatures i els grups l'usuari final pot rebre un arxiu *iCalendar* al seu correu, el qual podrà importar fàcilment a la seva plataforma de calendaris. Però no cal oblidar que aquest projecte únicament és un pont per accions futures, com ara la incorporació a la pàgina web d'un algoritme semblant al utilitzat en ETSEIB Calendar, on a partir d'*inputs* senzills es pugui simular el calendari final (actualment el portal d'assignatures i horaris ja ho realitza), calcular els possibles solapaments de pràctiques (ETSEIB Calendar ha prescindit de calcular els solapaments de pràctiques) i crear l'arxiu contenidor de les pràctiques de laboratori.

Treballar amb la plataforma Kivy ha facilitat molt les coses, però també les ha limitat, ja que en molts casos no es podia realitzar de forma senzilla allò que a primera vista es podria realitzar de forma senzilla en llenguatge html per exemple. El fet de no poder paquetitzar l'aplicació per més plataformes de forma trivial ha sigut un dels punts més febles de Kivy, ja que per tal de crear un executable de Windows o un debian per Linux calia instal·lar molts complements i era una operació força costosa. Finalment es va decidir centrar-se en Android ja que no es disposava del temps necessari per expandir-se a altres plataformes. Tal i com s'ha parlat al punt anterior, una possible expansió del projecte seria la conversió a altres plataformes o la incorporació a un prototip de web de l'escola. A més a més per indisposició de temps no s'ha pogut treballar el control estadístic de la plataforma, que seria interessant per temes de gestió dels usuaris (quines assignatures trien, en quin períodes de l'any es utilitzada l'aplicació...). La majoria d'objectius del projecte han estat assolits de manera satisfactòria, ja que alguns com ara treballar en moltes plataformes necessitaven més temps de desenvolupament i altres com treballar en games de colors positives/negatives han sigut solucionades amb la incorporació d'una gama de colors intermitja que permet la visibilitat en ambients de diferents lluminositats.

Agraïments

M'agradaria començar agraint el suport rebut per part del meu tutor Lluís Solano, que des del primer moment va veure amb bons ulls la idea de desenvolupar una eina que ajudés als estudiants de l'escola. A més a més aportant idees sobre eines per desenvolupar-la em va fer descobrir Kivy, columna vertebral d'ETSEIB Calendar. Aquesta projecte no seria possible sense la base de dades proveïda per els serveis informàtics de l'ETSEIB, i m'agradaria agrair l'ajuda a Jordi Rodríguez a la hora de personalitzar la bolcada de dades. A més a més, als meus amics i família, que en cap moment van dubtar en que el projecte sortiria endavant.

Bibliografia

Referències bibliogràfiques

- [1] Jordi Rodriguez Vila. Base de dades pràctiques ETSEIB [jordi.rodriquez-vila@upc.edu], 14/02/2016]
- [2] ICalendar. Documentació sobre els arxius iCalendar [<http://icalendar.org/>], 17/02/2016]
- [3] Kivy. Documentació sobre la plataforma Kivy [<https://kivy.org/docs/>], 17/02/2016]
- [4] StackOverFlow. Fòrum de consulta de programadors [<http://www.stackoverflow.com/>], 20/02/2016]
- [5] Youtube. Tutorials de desenvolupament Kivy [<https://www.youtube.com/user/sentdex>], 22/02/2016]
- [6] Wikipedia. Història de Python. [https://es.wikipedia.org/wiki/Historia_de_Python], 25/04/2016]
- [7] Kanzaki. Informació sobre el funcionament de les notificacions d'iCalendar [<http://www.kanzaki.com/docs/ical/valarm.html>], 12/04/2016]
- [8] JsonFormatter. Donar format a un document json. [<https://jsonformatter.curiousconcept.com/>], 25/04/2016]
- [9] Inclem. Com paquetitzar l'aplicació Kivy [http://inclem.net/2014/01/12/kivy-crash-course/2_building_an_android_apk/], 20/05/2016]
- [10] Planetb. Donar format Python per inserir a Word [<http://www.planetb.ca/syntax-highlight-word>], 01/06/2016]
- [11] Gmail Meter. Anàlisi estadístic de les tramitacions de calendaris [<http://www.gmailmeter.com/>], 20/08/2016]

Importància de les referències

En quant a importància de la bibliografia, cal destacar les següents:

- [1] Jordi Rodriguez Vila. Base de dades pràctiques ETSEIB [jordi.rodriguez-vila@upc.edu ,14/02/2016] → document d'on s'extreuen totes les dades de les pràctiques de laboratori.
- [2] ICalendar. Documentació sobre els arxius iCalendar [<http://icalendar.org/>, 17/02/2016]→ Arquitectura dels documents iCalendar i com treballar-los a Python.
- [3] Kivy. Documentació sobre la plataforma Kivy [<https://kivy.org/docs/>, 17/02/2016]→ Arquitectura de la plataforma Kivy i documentació sobre tots els elements que es poden assolir.
- [5] Youtube. Tutorials de desenvolupament Kivy [<https://www.youtube.com/user/sentdex>, 22/02/2016]→ Tutorials sobre com crear aplicacions a Kivy.
- [9] Inclem. Com paquetitzar l'aplicació Kivy [http://inclem.net/2014/01/12/kivy-crash-course/2_building_an_android_apk/, 20/05/2016]→ Documentació sobre la paquetització de les Kivy Apps.

SUMARI ANNEX

Llistat de figures annex	2
A. Explicació document Json.....	3
B. Exemple amb comentaris arxiu iCalendar	7
C. Script arxicsmulti.py.....	8
D. Script filtro.py	10
E. Script main.py	11
F. Script Kivy language.....	19
G. Captures de pantalla de la primera versió d'ETSEIB Calendar	27

Llistat de figures annex

<i>Figura A.1: Captura del primer nivell del document json</i>	<i>3</i>
<i>Figura A.2: Captura del GETI desplegat al document json.....</i>	<i>4</i>
<i>Figura A.3: Captura d'un exemple d'assignatura del document json</i>	<i>4</i>
<i>Figura A.4: Captura d'un exemple d'assignatura amb grups del document json</i>	<i>5</i>
<i>Figura A.5: Captura d'un exemple de sessions de laboratori d'una assignatura al document json.....</i>	<i>6</i>
<i>Figura G.1: Captura de la primera versió de pantalla inicial d'ETSEIB Calendar</i>	<i>27</i>
<i>Figura G.2: Captura de la primera pantalla de selecció d'assignatures de la primera versió d'ETSEIB Calendar.....</i>	<i>27</i>
<i>Figura G.3: Captura de la primera pantalla de selecció d'assignatures (amb seleccions) de la primera versió d'ETSEIB Calendar</i>	<i>28</i>
<i>Figura G.4: Captura de la segona pantalla de selecció d'assignatures de la primera versió d'ETSEIB Calendar.....</i>	<i>28</i>
<i>Figura G.5: Captura de la segona pantalla de selecció d'assignatures (amb seleccions) de la primera versió d'ETSEIB Calendar</i>	<i>29</i>
<i>Figura G.6: Captura de la pantalla de selecció de grups de la primera versió d'ETSEIB Calendar</i>	<i>29</i>
<i>Figura G.7: Captura de pantalla de la pantalla de selecció de grups de la primera versió d'ETSEIB Calendar amb un string incorrecte.....</i>	<i>30</i>
<i>Figura G.8: Captura de pantalla del popup error a la pantalla de selecció de grups de la primera versió d'ETSEIB Calendar</i>	<i>30</i>

A. Explicació document Json

Explicació del document json a través d'un cas pràctic de selecció de l'assignatura Control Automàtic (codi: 240172) grup 44.

Al accedir a la base de dades proveïda per els serveis informàtics de l'ETSEIB el resultat és una llista inicial que emmagatzema la totalitat de la informació de les pràctiques de laboratori. A la següent imatge es pot veure una captura dels elements que es troben formant part d'aquesta llista:

```
[
  {
    "degree": "GETI",
    "degreeName": "Grau en Enginyeria en Tecnologies Industrials",
    "subjects": [
    ],
  },
  {
    "degree": "GEQ",
    "degreeName": "Grau en Enginyeria Química",
    "subjects": [
    ],
  },
  {
    "degree": "GEM",
    "degreeName": "Grau en Enginyeria de Materials",
    "subjects": [
    ],
  }
]
```

Figura A.1: Captura del primer nivell del document json

Com es pot veure dins la llista trobem distribuïts 3 diccionaris, un per cada grau que es cursa a escola. Degut a que el projecte s'ha decidit que es centrarà únicament en el GETI es prescindirà de tota la informació referent als graus de química i materials. Per accedir a la totalitat de les dades de les pràctiques del GETI caldrà seleccionar l'element 0 de la llista que engloba el document json.


```
[
  {
    "degree": "GETI",
    "degreeName": "Grau en Enginyeria en Tecnologies Industrials",
    "subjects": [
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { }
    ]
  },
  { },
  { }
]
```

Figura A.2: Captura del GETI desplegat al document json

Aquesta llista esta composta per diferents diccionaris que emmagatzemen totes les pràctiques de laboratori de les assignatures que disposen de pràctiques (13 en total durant tot el grau). Com s'ha parlat anteriorment s'accedirà a Control Automàtic:

```
[
  {
    "degree": "GETI",
    "degreeName": "Grau en Enginyeria en Tecnologies Industrials",
    "subjects": [
      {
        "code": "240172",
        "nameCAT": "Control Automàtic",
        "semester": "7",
        "groups": [
          { }
        ]
      },
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { },
      { }
    ]
  },
  { },
  { }
]
```

Figura A.3: Captura d'un exemple d'assignatura del document json

Els noms estan en format html, per tant caldrà donar-li format utf-8 per veure els accents. Dintre del diccionari a la clau *groups* es troben integrats la totalitat de grups que corresponen a aquesta assignatura, en format de llista. Cal tenir en compte que quan un estudiant es matricula en una assignatura que disposa de pràctiques de laboratori haurà d'escollir un grup múltiple de 10 per la teoria, i un altre per les sessions de laboratori (per exemple 11,12,13...). A la següent imatge es pot veure la sèrie de grups que formen part de l'assignatura de Control Automàtic (16 possibilitats de grup de pràctiques):

```
[
  {
    "degree": "GETI",
    "degreeName": "Grau en Enginyeria en Tecnologies Industrials",
    "subjects": [
      {
        "code": "240172",
        "nameCAT": "Control Automàtic",
        "semester": "7",
        "groups": [
          "44": [ ],
          "41": [ ],
          "43": [ ],
          "42": [ ],
          "32": [ ],
          "34": [ ],
          "31": [ ],
          "33": [ ],
          "22": [ ],
          "21": [ ],
          "24": [ ],
          "23": [ ],
          "13": [ ],
          "11": [ ],
          "12": [ ],
          "14": [ ]
        ]
      }
    ]
  },
]
```

Figura A.4: Captura d'un exemple d'assignatura amb grups del document json

D'entre totes les possibilitats seleccionarem la llista que correspon al grup 44, que s'obtindrà a través de la clau "44". Un cop seleccionat el grup s'obtindrà una llista de tants elements com pràctiques tingui la assignatura, en aquest cas 4:

```
{
  "code": "240172",
  "nameCAT": "Control Automàtic",
  "semester": "7",
  "groups": [
    {
      "44": [
        {
          "day": "13-04-2016",
          "startHour": "09:00:00",
          "endHour": "12:00:00",
          "classroom": "H-H 2.21"
        },
        {
          "day": "27-04-2016",
          "startHour": "09:00:00",
          "endHour": "12:00:00",
          "classroom": "H-H 2.21"
        },
        {
          "day": "11-05-2016",
          "startHour": "09:00:00",
          "endHour": "12:00:00",
          "classroom": "H-H 2.21"
        },
        {
          "day": "25-05-2016",
          "startHour": "09:00:00",
          "endHour": "12:00:00",
          "classroom": "H-H 2.21"
        }
      ]
    }
  ]
}
```

Figura A.5: Captura d'un exemple de sessions de laboratori d'una assignatura al document json

Per l'assignatura de Control automàtic, el grup 44 consta de 4 pràctiques. La primera d'aquestes és el dia 13-04-2016 de 9:00 a 12:00 i a la classe H-H 2.21. La segona, la tercera i la quarta tindran el mateix horari, la mateixa aula i es duran a terme els dies 27-04-2016, 11-05-2016 i 25-05-2016.

B. Exemple amb comentaris arxiu iCalendar

A continuació s'analitzarà un arxiu d'extensió .ics (iCalendar) que conté una pràctica del grup 11 de Termodinàmica. L'objectiu de l'aplicació ETSEIB Calendar es poder crear un que integri tots els camps automatitzats, amb totes les pràctiques que l'alumne seleccioni. Cada línia està comentada amb la seva finalitat.

BEGIN:VCALENDAR→ Inicialitza el calendari

BEGIN:VEVENT→ Inicialitza un esdeveniment del calendari

SUMMARY:Pràctica 1 de Termodinàmica→ Resum de l'esdeveniment

DTSTART;VALUE=DATE-TIME: 20160309T120000→ Inici de l'esdeveniment (any-mes-dia-hora)

DTEND;VALUE=DATE-TIME: 20160309T170000→ Final de l'esdeveniment (any-mes-dia-hora)

LOCATION:Etseib H-H 7.56→ Localització de l'esdeveniment

BEGIN:VALARM→ Inicialització de l'alarma

ACTION:EMAIL→ Acció que executarà l'alarma, en aquest cas s'enviarà un correu

DESCRIPTION:Demà Pràctica 1 de Termodinàmica→ Descripció que contindrà el correu que s'enviarà

TRIGGER:-PT1H→ Anterioritat a l'esdeveniment amb la que s'executarà l'alarma, en aquest cas 1 hora

END:VALARM→ Tancament de les especificacions de l'alarma

END:VEVENT→ Tancament de l'esdeveniment

END:VCALENDAR→ Tancament del calendari

C. Script arxicsmulti.py

```

1. # -*- coding: utf-8 -*-
2. from icalendar import Calendar, vCalAddress, vText
3. import pytz
4. from datetime import datetime, timedelta
5. import icalendar
6. import smtplib
7. from email.MIMEMultipart import MIMEMultipart
8. from email.MIMEText import MIMEText
9. from email.MIMEBase import MIMEBase
10. from email import encoders
11.
12. def calendar_creator(classe, inici, final, dia, assignatura, trigger, mail):
13.     cal = Calendar()
14.     cal.add('prodid', '-//Calendari Pràctiques ETSEIB//')
15.     cal.add('version', '2.0')
16.     e=[]
17.     for i in range(0, len(classe)):
18.         event=icalendar.Event()
19.         event['uid']=str(e.count(assignatura[i]))+assignatura[i]
20.         data=dia[i].split('-')
21.         alarm=icalendar.Alarm()
22.         day= int(data[0])
23.         mes=int(data[1])
24.         year=int(data[2])
25.         tini=inici[i].split(':')
26.         hini=int(tini[0])
27.         mini=int(tini[1])
28.         tfi=final[i].split(':')
29.         hfi=int(tfi[0])
30.         mfi=int(tfi[1])
31.         e.append(assignatura[i])
32.         desc="Pràctica "+" "+str(e.count(assignatura[i]))+" de "+
33.         assignatura[i].encode('utf-8')
34.         aula=classe[i]
35.         event.add('summary', desc)
36.         event.add('dtstart', datetime(year,mes,day,hini,mini,0,1))
37.         event.add('dtend', datetime(year,mes,day,hfi,mfi,0,1))
38.         event.add('dtstamp', datetime(year,mes,day,hini,mini,0,1))
39.         event['location'] = vText('Etseib '+aula)
40.         alarm.add('action','EMAIL')
41.         alarm.add('description', 'Tens la '+desc)
42.         alarm.add('trigger', timedelta(hours=-trigger))
43.         event.add_component(alarm)
44.         cal.add_component(event)
45.         fromaddr = "etseibcalendar@gmail.com"
46.         toaddr = mail.strip()
47.         msg = MIMEMultipart('alternative')
48.         msg['From'] = "ETSEIB Calendar <etseib.calendar@gmail.com>"
49.         msg['To'] = toaddr
50.         msg['Subject'] = u"Calendari Pràctiques ETSEIB"
51.         body = ""
52.         body += "<b>Instruccions per importar l'arxiu icalendar a Google Calendar:</b>"
53.         body += "<br>1: Descarrega l'arxiu adjunt del email.<br>"
54.         body += "2: Obre Google calendar en un ordinador (només es pot importar calendaris des d'un"
55.         body += "ordinador).<br>3: Adalt a la dreta, fes clic a configuració.<br>4: Obre la pestanya"
56.         body += "de Calendaris.<br>5: Fes clic a Importar calendaris.<br>6: Fes clic a selecciona ar"
57.         body += "xiu i escull l'arxiu que has descarregat anteriorment.<br>7: Escull a quin calendar"
58.         body += "i desitges importar-"
59.         body += "ho.<br>8: Fes clic a Importar.<br><br><b>Instruccions per importar l'arxiu icalenda

```

```

r a Outlook:</b><br>1: Descarrega l'arxiu adjunt del email.<br>2: Obre Outlook en u
n ordinador (només es pot importar calendaris des d'un ordinador).<br>3: Adalt a la
esquerra, fes clic a aplicacions.<br>4: Selecciona Calendari.<br>5: Fes clic a Agre
gar Calendaris.<br>6: Fes clic a "Desde un arxiu" i escull l'arxiu que has descarre
gat anteriorment.<br>7: Clica a Guardar.""
52.     msg.attach(MIMEText(body, "html"))
53.     filename = "etseibcalendarfile.ics"
54.     part = MIMEBase('application', 'octet-stream', name=filename)
55.     part.set_payload(cal.to_ical())
56.     encoders.encode_base64(part)
57.     part.add_header('Content-
Disposition', "attachment; filename= %s" % filename)
58.     msg.attach(part)
59.     server = smtplib.SMTP('smtp.gmail.com', 587)
60.     server.starttls()
61.     server.login(fromaddr, "etseibcalendar")
62.     text = msg.as_string()
63.     server.sendmail(fromaddr, toaddr, text)
64.     server.quit()
```

D. Script filtro.py

```

1. # -*- coding: utf-8 -*-
2. import json
3. import urllib
4. import HTMLParser
5. import arxicsmulti
6.
7. url="https://guiadocent.etseib.upc.edu/guiadocent/profile/default/action/exportExtraT
   imetablesJSON.php"
8. extr=urllib.urlopen(url)
9. data=json.loads(extr.read())
10.
11. def cerca(codi):
12.     grups=[]
13.     for e in range(0,len(data[0]["subjects"])):
14.         if data[0]["subjects"][e]["code"].encode("utf-8")==codi:
15.             for k in data[0]["subjects"][e]["groups"][0]:
16.                 grups.append(k.encode("utf-8"))
17.             grups.sort()
18.     return grups
19.
20. def dic_codis_signatures(n):
21.     d={}
22.     for e in range(0,len(data[n]['subjects'])):
23.         d[ data[n]["subjects"][e]["code"].encode("utf-
            8")] =HTMLParser.HTMLParser().unescape(data[0]["subjects"][e]["nameCAT"].encode('utf-
            8'))
24.     return d
25.
26. def cerca_horaris(codis,grups,trigger,mail):
27.     classe=[]
28.     inici=[]
29.     final=[]
30.     dia=[]
31.     ass=[]
32.     if len(codis)==len(grups):
33.         for n in range(0,len(grups)):
34.             for e in range(0,len(data[0]["subjects"])):
35.                 if data[0]["subjects"][e]["code"].encode("utf-8")==codis[n]:
36.                     T=e
37.                     break
38.             else:
39.                 pass
40.             for i in range(0,len(data[0]["subjects"][e]["groups"][0][grups[n]])):
41.                 try:
42.                     ass.append(HTMLParser.HTMLParser().unescape(data[0]["subjects"][
                        e]["nameCAT"].encode('utf-8'))))
43.                     classe.append(data[0]["subjects"][e]["groups"][0][grups[n]][i][ '
                        classroom'].encode("utf-8"))
44.                     inici.append(data[0]["subjects"][e]["groups"][0][grups[n]][i][ 's
                        tartHour'].encode("utf-8"))
45.                     final.append(data[0]["subjects"][e]["groups"][0][grups[n]][i][ 'e
                        ndHour'].encode("utf-8"))
46.                     dia.append(data[0]["subjects"][e]["groups"][0][grups[n]][i][ 'day
                        '].encode("utf-8"))
47.                 except:
48.                     pass
49.     return arxicsmulti.calendar_creator(classe,inici,final,dia,ass,trigger,mail)

```

E. Script main.py

```

1.  # -*- coding: utf-8 -*-
2.  from kivy.app import App
3.  from kivy.lang import Builder
4.  from kivy.uix.screenmanager import ScreenManager, Screen, SwapTransition
5.  from kivy.uix.textinput import TextInput
6.  from kivy.uix.label import Label
7.  from kivy.uix.gridlayout import GridLayout
8.  from kivy.uix.button import Button
9.  from kivy.properties import DictProperty
10. from kivy.properties import StringProperty
11. from smtplib import SMTPRecipientsRefused
12. from kivy.core.window import Window
13. from kivy.uix.popup import Popup
14. import socket
15. Window.softinput_mode = 'below_target'
16. try:
17.     import filtro
18.     internet=True
19. except:
20.     internet=False
21.
22.
23. Builder.load_string('''
24. #import FadeTransition kivy.uix.screenmanager.FadeTransition
25. <Root>:
26.     transition: FadeTransition()
27.     Welcome:
28.         name: 'welcome'
29.     MainScreen:
30.         name: 'main'
31.     SecondScreen:
32.         name: 'second'
33.     AnotherScreen:
34.         name: 'another'
35. <Image>
36.     keep_ratio: True
37.     allow_stretch: True
38.     opacity: 1
39. <Popup>
40.     size_hint: 0.7, 0.35
41.     background_color: 0, 0, 0, 0
42. <CheckBox>
43.     background_checkbox_normal: 'unchecked.png'
44.     background_checkbox_down: 'checked.png'
45. <Label>
46.     markup: True
47.     font_size: sp(16)
48. <TextInput>
49.     padding: [12,( self.height - self.line_height ) / 2]
50.
51. <Welcome>:
52.     canvas.before:
53.         Color:
54.             rgba: 0.5, 0.5, 0.5, 0.8
55.         BorderImage:
56.             border: 10, 10, 10, 10
57.             source: 'etseib.jpg'
58.             pos: self.pos

```



```

59.         size: self.size
60.     FloatLayout:
61.
62.
63.         Image:
64.             source: 'escruiocorreu.png'
65.             size_hint_x: 0.16
66.             pos_hint: {'right':0.395,'center_y':0.6}
67.         Image:
68.             source: 'best.png'
69.             size_hint_x: 0.85
70.             pos_hint: {'center_x':0.5,'center_y':0.375}
71.
72.         Spinner:
73.             #font_size: sp(15)
74.             text: "Quan vols rebre l'avís?"
75.             values:('24 hores abans','8 hores abans','2 hores abans','1 hora abans'
76.             )
77.             pos_hint: {'center_x':0.5,'center_y':0.19}
78.             size_hint: 0.2, 0.0575
79.             on_text: root.emmagatzema_trigger(self.text)
80.
81.         Button:
82.             text: "[b]Surt[/b]"
83.             background_color: 0.34, 1.83, 2, 1
84.             on_release: root.close()
85.             size_hint: 0.5, 0.1111111111
86.             pos_hint: {'center_x':0.25,'center_y':0.06}
87.
88.         Button:
89.             text: '[b]Següent[/b]'
90.             background_color: 6,1.5,0,1
91.             on_release:root.ready()
92.             size_hint: 0.5, 0.1111111111
93.             pos_hint: {'center_x':0.75,'center_y': 0.06}
94.
95.         TextInput:
96.             multiline: False
97.             input_type: 'mail'
98.             size_hint: 0.2, 0.0595
99.             #font_size: sp(15)
100.            pos_hint: {'center_x':0.5,'center_y':0.6}
101.            hint_text:'exemple@gmail.com'
102.            on_text: root.emmagatzema_email(self.text)
103.
104.
105.         Image:
106.             source: 'calendar.png'
107.             size_hint: 0.52, 0.375
108.             pos_hint: {'center_x': 0.5, 'center_y': 0.83}
109.
110.
111.         Button:
112.             background_down:'help2.png'
113.             background_normal: 'help.png'
114.             size_hint: 0.1,0.15
115.             pos_hint: {'center_x': 0.85, 'center_y': 0.85}
116.             on_release: root.info()
117.
118.
119.

```

```

120.
121.
122. <MainScreen>:
123.     canvas.before:
124.         Color:
125.             rgba: 1, 1, 1, 0.8
126.         BorderImage:
127.             border: 10, 10, 10, 10
128.             source: 'hd.jpg'
129.             pos: self.pos
130.             size: self.size
131.         GridLayout:
132.             cols: 2
133.             Label:
134.                 text: "[b]Selecciona Assignatures (1/2)[/b]"
135.                 font_size: sp(17)
136.             Label:
137.                 text: " "
138.             CheckBox:
139.                 on_active:root.emmagatzema('240031',self.active)
140.
141.             Label:
142.                 text: "[b]Electromagnetisme[/b]"
143.             CheckBox:
144.                 on_active:root.emmagatzema('240033',self.active)
145.             Label:
146.                 text: "[b]Materials[/b]"
147.             CheckBox:
148.                 on_active:root.emmagatzema('240052',self.active)
149.             Label:
150.                 text: "[b]Termodinàmica[/b]"
151.             CheckBox:
152.                 on_active:root.emmagatzema('240053',self.active)
153.             Label:
154.                 text: "[b]Electrotècnia[/b]"
155.             CheckBox:
156.                 on_active:root.emmagatzema('240054',self.active)
157.             Label:
158.                 text: "[b]Mecànica dels Medis Continus[/b]"
159.             CheckBox:
160.                 on_active:root.emmagatzema('240061',self.active)
161.             Label:
162.                 text: "[b]Mecànica de Fluids[/b]"
163.             CheckBox:
164.                 on_active:root.emmagatzema('240063',self.active)
165.             Label:
166.                 text: "[b]Resistència de Materials[/b]"
167.
168.             Button:
169.                 text: "[b]Inici[/b]"
170.                 background_color: 0.34, 1.83, 2, 1
171.                 on_release:
172.                     root.parent.current='welcome'
173.
174.             Button:
175.                 text: "[b]Següent[/b]"
176.                 font_size:
177.                 background_color: 6,1.5,0,1
178.                 on_release:
179.                     root.parent.current='second'
180.
181.

```

```

182.
183. <SecondScreen>:
184.     canvas.before:
185.         Color:
186.             rgba: 1, 1, 1, 0.7
187.         BorderImage:
188.             border: 10, 10, 10, 10
189.             source: 'hd.jpg'
190.             pos: self.pos
191.             size: self.size
192.         GridLayout:
193.             cols: 2
194.             Label:
195.                 text: "[b]Selecciona Assignatures (2/2)[/b]"
196.                 font_size: sp(17)
197.             Label:
198.                 text: " "
199.             CheckBox:
200.                 on_active:root.emmagatzema('240072',self.active)
201.             Label:
202.                 text: "[b]Electrònica[/b]"
203.             CheckBox:
204.                 on_active:root.emmagatzema('240073',self.active)
205.             Label:
206.                 text: "[b]Sistemes de Fabricació [/b]"
207.             CheckBox:
208.                 on_active:root.emmagatzema('240151',self.active)
209.             Label:
210.                 text: "[b]Tecnologia i Selecció de Materials[/b]"
211.             CheckBox:
212.                 on_active:root.emmagatzema('240161',self.active)
213.             Label:
214.                 text: "[b]Màquines Elèctriques[/b]"
215.             CheckBox:
216.                 on_active:root.emmagatzema('240171',self.active)
217.             Label:
218.                 text: "[b]Termotècnia[/b]"
219.             CheckBox:
220.                 on_active:root.emmagatzema('240172',self.active)
221.             Label:
222.                 text: "[b]Control Automàtic[/b]"
223.             Label:
224.                 text: ""
225.             Label:
226.                 text: ""
227.             Button:
228.                 text: "[b]Anterior[/b]"
229.                 background_color: 0.34, 1.83, 2, 1
230.                 on_release:
231.                     root.parent.current='main'
232.             Button:
233.                 text: "[b]Següent[/b]"
234.                 background_color: 6,1.5,0,1
235.                 on_release:
236.                     root.ready()
237. <AnotherScreen>:
238.     canvas.before:
239.         Color:
240.             rgba: 1, 1, 1, 0.8
241.         BorderImage:
242.             border: 10, 10, 10, 10
243.             source: 'hd.jpg'

```

```

244.         pos: self.pos
245.         size: self.size
246.     GridLayout:
247.         id: container
248.         cols: 2
249.
250.
251.     '''
252.
253.
254.
255. class Welcome(Screen):
256.     def __init__(self, **kw):
257.         super(Welcome, self).__init__(**kw)
258.         self.a = App.get_running_app()
259.     def emmagatzema_email(self, text_inserit):
260.         self.a.email= text_inserit
261.
262.     def info(self):
263.         content=Label(text='[b]Desenvolupador: Alejandro Piquer Robledo (Contacte:
264. etseib.calendar@gmail.com)[/b]',font_size='15sp')
265.         popup = Popup(title='Informació Kivy App', content=content)
266.         popup.open()
267.         content.bind(on_press=popup.dismiss)
268.
269.     def emmagatzema_trigger(self,seleccio):
270.         self.a.trigger=seleccio[0:2]
271.
272.     def close(self):
273.         return App.get_running_app().stop()
274.
275.     def ready(self):
276.         if self.a.trigger=="default" or self.a.email=="":
277.             content=Label(text='[b]Cal que seleccionis tots els camps[/b]',font_siz
278. e='15sp')
279.             popup = Popup(title='Error',content=content)
280.             popup.open()
281.             content.bind(on_press=popup.dismiss)
282.         else:
283.             if internet==False:
284.                 content=Label(text="[b]Revisa la teva connexió a Internet i reinici
285. a l'app[/b]",font_size='15sp')
286.                 popup = Popup(title='Informació',content=content)
287.                 popup.open()
288.             else:
289.                 self.parent.current='main'
290.
291. class MainScreen(Screen):
292.     def __init__(self, **kw):
293.         super(MainScreen, self).__init__(**kw)
294.         self.a = App.get_running_app()
295.
296.     def emmagatzema(self, n, value):
297.         self.a.big_dict[n] = value
298.
299. class SecondScreen(Screen):
300.     def __init__(self, **kw):
301.         super(SecondScreen, self).__init__(**kw)
302.         self.a = App.get_running_app()

```

```

303.     def emmagatzema(self, n, value):
304.         self.a.big_dict[n] = value
305.
306.     def ready(self):
307.         checkbox = False
308.         for codi_assignatura in App.get_running_app().big_dict.values():
309.             if codi_assignatura == False:
310.                 pass
311.             else:
312.                 checkbox = True
313.                 break
314.         if checkbox == False:
315.             content = Label(text='[b]Cal que seleccionis alguna assignatura[/b]', font
_size='15sp')
316.             popup = Popup(title='Error', content=content)
317.             popup.open()
318.             content.bind(on_press=popup.dismiss)
319.         else:
320.             try:
321.                 self.parent.current = 'another'
322.             except:
323.                 content = Label(text='[b]Revisa la teva connexió a Internet[/b]', font
_size='15sp')
324.                 popup = Popup(title='Informació', content=content)
325.                 popup.open()
326.
327.
328.
329.     class AnotherScreen(Screen):
330.         def on_pre_enter(self, *args):
331.             self.ids.container.clear_widgets()
332.             dic_signatures = filtro.dic_codis_signatures(0)
333.             a = App.get_running_app()
334.             text_inputs = {}
335.             keys = []
336.             self.ids.container.add_widget(Label(text="[b]Assignatura[/b]", font_size='17
sp'))
337.             self.ids.container.add_widget(Label(text="[b]Escriu el grup[/b]", font_size=
'17sp'))
338.         def emmagatzema_textinputs(self, seleccio):
339.             text_inputs[self.id] = seleccio
340.
341.         for e in a.big_dict:
342.             keys.append(e)
343.             keys.sort()
344.         for codi in keys:
345.             if a.big_dict[codi] == True:
346.                 e = codi
347.                 self.ids.container.add_widget(Label(text='[b]+' + codi + '-
'+dic_signatures[codi].encode("utf-8")+ '[/b]'))
348.                 suggeriment_grups = str(filtro.cerca(codi.encode("utf-8")))[1:-
1].replace("'", "")
349.                 self.codi = TextInput(hint_text=suggeriment_grups, multiline=False, id=
codi, write_tab=False, input_type='number', input_filter='int', background_color=[1,1,1,0
.8], font_size='15sp')
350.                 self.codi.bind(text=emmagatzema_textinputs)
351.                 self.ids.container.add_widget(self.codi)
352.
353.     def change1(Screen):
354.         self.parent.current = 'welcome'
355.     def change2(Screen):
356.         self.parent.current = 'second'

```

```

357.
358.     def executa(self):
359.         ass=[]
360.         grup=[]
361.         error=False
362.         def popitup():
363.             content=Label(text='[b]Operació realitzada amb èxit, revisa el teu e
mail[/b]',font_size='17sp')
364.             popup = Popup(title='Operació realitzada',content=content)
365.             popup.open()
366.             content.bind(on_press=popup.dismiss)
367.         for e in text_inputs:
368.             try:
369.                 if text_inputs[e] in filtro.cerca(e.encode("utf-8")):
370.                     x=float(text_inputs[e])
371.                     ass.append(e)
372.                     grup.append(text_inputs[e].encode("utf-8"))
373.             else:
374.                 raise ValueError('No existeix la teva selecció')
375.             except:
376.                 content=Label(text='[b]Insereix grups vàlids[/b]')
377.                 popup = Popup(title='Error de grup',content=content,size_hint=(
0.7, 0.5),background_color=(0, 0, 0),font_size='15sp')
378.                 popup.open()
379.                 content.bind(on_press=popup.dismiss)
380.                 ass=[]
381.                 grup=[]
382.                 error=True
383.                 break
384.         if error==False:
385.             try:
386.                 return filtro.cerca_horaris(ass,grup,int(a.trigger),a.email),po
pitup()
387.             except SMTPRecipientsRefused:
388.                 content=Button(text='[b]El email inserit no és vàlid[/b]',on_re
lease=change1,size_hint= (0.6, 0.3),font_size='15sp')
389.                 popup = Popup(title='Error de email',content=content)
390.                 popup.open()
391.                 content.bind(on_press=popup.dismiss)
392.
393.
394.         b1=Button(text='[b]Anterior[/b]',background_color=[0.34, 1.83, 2, 1], size_hint
_y= None,height= int(Window.height)/8.8 )
395.         self.ids.container.add_widget(b1)
396.         b2=Button(text='[b]Executa[/b]',background_color=[6,1.5,0,1],size_hint_y= None,
height= int(Window.height)/8.8)
397.         self.ids.container.add_widget(b2)
398.         b1.bind(on_release=change2)
399.         b2.bind(on_release=executa)
400.
401.
402.
403.     class Root(ScreenManager):
404.         pass
405.
406.     class SimpleKivy(App):
407.         big_dict = DictProperty({'240161': False, '240061': False, '240171': False, '24
0151': False, '240063': False, '240073': False, '240072': False, '240031': False, '24
0033': False, '240054': False, '240053': False, '240052': False, '240172': False})
408.     )
409.         email=StringProperty("default")
410.         trigger=StringProperty("default")

```

```
411.  
412.     def build(self):  
413.         return Root()  
414. SimpleKivy().run()
```

F. Script Kivy language

A continuació es pot veure la totalitat del kivy language necessari per desenvolupar ETSEIB Calendar:

```
#:import FadeTransition kivy.uix.screenmanager.FadeTransition
```

```
<Root>:
```

```
    transition: FadeTransition()
```

```
    Welcome:
```

```
        name: 'welcome'
```

```
    MainScreen:
```

```
        name: 'main'
```

```
    SecondScreen:
```

```
        name: 'second'
```

```
    AnotherScreen:
```

```
        name: 'another'
```

```
<Image>
```

```
    keep_ratio: True
```

```
    allow_stretch: True
```

```
    opacity: 1
```

```
<Popup>
```

```
    size_hint: 0.7, 0.35
```

```
    background_color: 0, 0, 0, 0
```

```
<CheckBox>
```

```
    background_checkbox_normal: 'unchecked.png'
```

```
    background_checkbox_down: 'checked.png'
```

```
<Label>
```

```
    markup: True
```

```
    font_size: sp(16)
```

```
<TextInput>
```

```
    padding: [12,( self.height - self.line_height ) / 2]
```


<Welcome>:

canvas.before:

Color:

rgba: 0.5, 0.5, 0.5, 0.8

BorderImage:

border: 10, 10, 10, 10

source: 'etseib.jpg'

pos: self.pos

size: self.size

FloatLayout:

Image:

source: 'escricuorreu.png'

size_hint_x: 0.16

pos_hint: {'right':0.395,'center_y':0.6}

Image:

source: 'best.png'

size_hint_x: 0.85

pos_hint: {'center_x':0.5,'center_y':0.375}

Spinner:

#font_size: sp(15)

text: "Quan vols rebre l'avís?"

values:('24 hores abans','8 hores abans','2 hores abans','1 hora abans')

pos_hint: {'center_x':0.5,'center_y':0.19}

size_hint: 0.2, 0.0575

on_text: root.emmagatzema_trigger(self.text)

Button:

```
text: "[b]Surt[/b]"
background_color: 0.34, 1.83, 2, 1
on_release: root.close()
size_hint: 0.5, 0.1111111111
pos_hint: {'center_x':0.25,'center_y':0.06}
```

Button:

```
text: '[b]Següent[/b]'
background_color: 6,1.5,0,1
on_release:root.ready()
size_hint: 0.5, 0.1111111111
pos_hint: {'center_x':0.75,'center_y': 0.06}
```

TextInput:

```
multiline: False
input_type: 'mail'
size_hint: 0.2, 0.0595
pos_hint: {'center_x':0.5,'center_y':0.6}
hint_text:'exemple@gmail.com'
on_text: root.emmagatzema_email(self.text)
```

Image:

```
source: 'calendar.png'
size_hint: 0.52, 0.375
pos_hint: {'center_x': 0.5, 'center_y': 0.83}
```

Button:

```
background_down: 'help2.png'
background_normal: 'help.png'
size_hint: 0.1, 0.15
pos_hint: {'center_x': 0.85, 'center_y': 0.85}
on_release: root.info()
```

<MainScreen>:

canvas.before:

Color:

```
rgba: 1, 1, 1, 0.8
```

BorderImage:

```
border: 10, 10, 10, 10
source: 'hd.jpg'
pos: self.pos
size: self.size
```

GridLayout:

cols: 2

Label:

```
text: "[b]Selecciona Assignatures (1/2)[/b]"
font_size: sp(17)
```

Label:

```
text: " "
```

CheckBox:

```
on_active: root.emmagatzema('240031', self.active)
```

Label:

text: "[b]Electromagnetisme[/b]"

CheckBox:

on_active:root.emmagatzema('240033',self.active)

Label:

text: "[b]Materials[/b]"

CheckBox:

on_active:root.emmagatzema('240052',self.active)

Label:

text: "[b]Termodinàmica[/b]"

CheckBox:

on_active:root.emmagatzema('240053',self.active)

Label:

text: "[b]Electrotècnia[/b]"

CheckBox:

on_active:root.emmagatzema('240054',self.active)

Label:

text: "[b]Mecànica dels Medis Continus[/b]"

CheckBox:

on_active:root.emmagatzema('240061',self.active)

Label:

text: "[b]Mecànica de Fluids[/b]"

CheckBox:

on_active:root.emmagatzema('240063',self.active)

Label:

text: "[b]Resistència de Materials[/b]"

Button:

text: "[b]Inici[/b]"

```
background_color: 0.34, 1.83, 2, 1  
on_release: root.parent.current='welcome'
```

Button:

```
text: "[b]Següent[/b]"  
font_size:  
background_color: 6,1.5,0,1  
on_release: root.parent.current='second'
```

<SecondScreen>:

canvas.before:

Color:

```
rgba: 1, 1, 1, 0.7
```

BorderImage:

```
border: 10, 10, 10, 10  
source: 'hd.jpg'  
pos: self.pos  
size: self.size
```

GridLayout:

cols: 2

Label:

```
text: "[b]Selecciona Assignatures (2/2)[/b]"  
font_size: sp(17)
```

Label:

```
text: " "
```

CheckBox:

```
on_active:root.emmagatzema('240072',self.active)
```

Label:

```
text: "[b]Electrònica[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240073',self.active)
```

Label:

```
text: "[b]Sistemes de Fabricació[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240151',self.active)
```

Label:

```
text: "[b]Tecnologia i Selecció de Materials[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240161',self.active)
```

Label:

```
text: "[b]Màquines Elèctriques[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240171',self.active)
```

Label:

```
text: "[b]Termotècnia[/b]"
```

CheckBox:

```
on_active:root.emmagatzema('240172',self.active)
```

Label:

```
text: "[b]Control Automàtic[/b]"
```

Label:

```
text: ""
```

Label:

```
text: ""
```

Button:

```
text: "[b]Anterior[/b]"
```

```
background_color: 0.34, 1.83, 2, 1
```

```
on_release:
```

```
root.parent.transition.direction = 'right'
```

```
root.parent.current='main'
```

Button:

```
text: "[b]Següent[/b]"
```

```
background_color: 6,1.5,0,1
```

```
on_release:
```

```
root.ready()
```

<AnotherScreen>:

canvas.before:

Color:

```
rgba: 1, 1, 1, 0.8
```

BorderImage:

```
border: 10, 10, 10, 10
```

```
source: 'hd.jpg'
```

```
pos: self.pos
```

```
size: self.size
```

GridLayout:

```
id: container
```

```
cols: 2
```

G. Captures de pantalla de la primera versió d'ETSEIB Calendar

A continuació es poden veure algunes captures de pantalla de la primera versió totalment funcional d'ETSEIB Calendar:

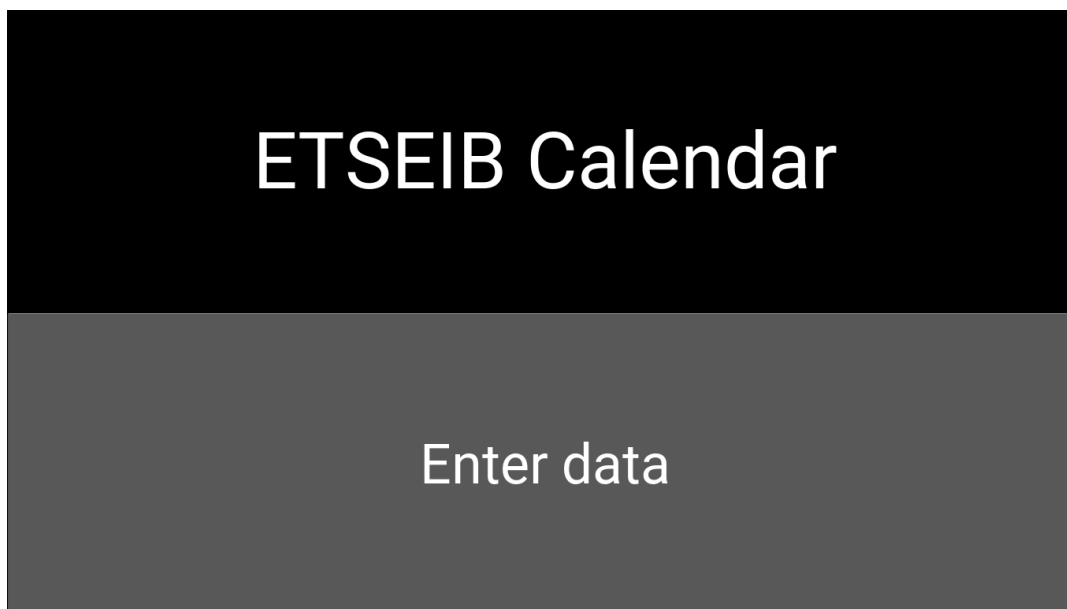


Figura G.1: Captura de la primera versió de pantalla inicial d'ETSEIB Calendar

Selecció al botó Enter data s'avança a la següent pantalla :

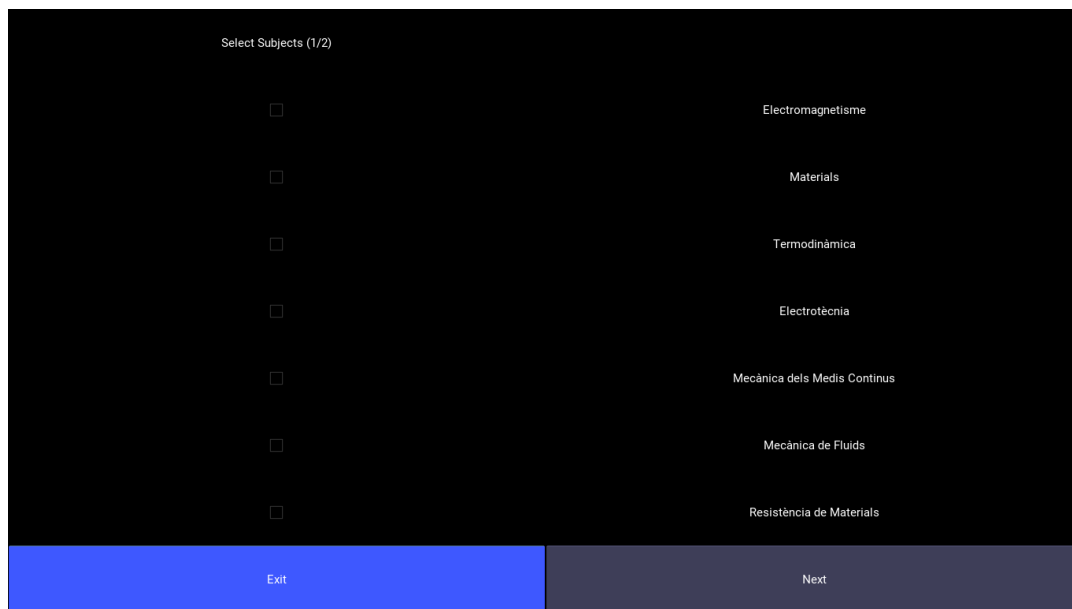


Figura G.2: Captura de la primera pantalla de selecció d'assignatures de la primera versió d'ETSEIB Calendar

Es pot veure com les *checkboxes* no són visualment molt atractives, es seleccionen algunes assignatures:

Subject	Selected
Electromagnetisme	<input checked="" type="checkbox"/>
Materials	<input type="checkbox"/>
Termodinàmica	<input type="checkbox"/>
Electrotècnia	<input type="checkbox"/>
Mecànica dels Medis Continus	<input checked="" type="checkbox"/>
Mecànica de Fluids	<input checked="" type="checkbox"/>
Resistència de Materials	<input type="checkbox"/>

Exit Next

Figura G.3: Captura de la primera pantalla de selecció d'assignatures (amb seleccions) de la primera versió d'ETSEIB Calendar

Pitjant a “Next” es pot seleccionar entre les altres assignatures:

Subject	Selected
Electrònica	<input type="checkbox"/>
Sistemes de Fabricació	<input type="checkbox"/>
Tecnologia i Selecció de Materials	<input type="checkbox"/>
Màquines Elèctriques	<input type="checkbox"/>
Termotècnia	<input type="checkbox"/>
Control Automàtic	<input type="checkbox"/>

Previous Next

Figura G.4: Captura de la segona pantalla de selecció d'assignatures de la primera versió d'ETSEIB Calendar

Seleccionem algunes altres:

Select Subjects (2/2)	
<input type="checkbox"/>	Electrònica
<input checked="" type="checkbox"/>	Sistemes de Fabricació
<input type="checkbox"/>	Tecnologia i Selecció de Materials
<input type="checkbox"/>	Màquines Elèctriques
<input checked="" type="checkbox"/>	Termotècnia
<input type="checkbox"/>	Control Automàtic

Previous
Next

Figura G.5: Captura de la segona pantalla de selecció d'assignatures (amb seleccions) de la primera versió d'ETSEIB Calendar

Donarà com a resultat la següent pantalla de selecció de grups:

Assignatura	Escriu el grup
240031-Electromagnetisme	
240054-Mecànica dels Medis Continus	
240061-Mecànica de Fluids	
240073-Sistemes de Fabricació	
240171-Termotècnia	

Exit
Run

Figura G.6: Captura de la pantalla de selecció de grups de la primera versió d'ETSEIB Calendar

Es pot veure amb claredat que els botons de l'última fila no respecten la mida de les anteriors pantalles, en aquesta primera fase tot era proporcional e igual . A més a més no es mostra la sèrie de grups que són seleccionables i el teclat és alfanumèric, permetent inserir *strings*, com es pot veure a la següent imatge:

Assignatura	Escriu el grup
240031-Electromagnetisme	12
240033-Materials	12
240053-Electrotècnia	31
240061-Mecànica de Fluids	12
240161-Màquines Elèctriques	ara
Exit	Run

Figura G.7: Captura de pantalla de la pantalla de selecció de grups de la primera versió d'ETSEIB Calendar amb un string incorrecte

Que donarà com a resultat al pitjar el botó “Run” el següent error:

Assignatura	Escriu el grup
240031-Electromagnetisme	12
240033-Materials	
240053-Electrotècnia	
240061-Mecànica de Fluids	
240161-Màquines Elèctriques	
Exit	Run

Error

Insert Valid Data

Figura G.8: Captura de pantalla del popup error a la pantalla de selecció de grups de la primera versió d'ETSEIB Calendar

En cas d'inserir grups de forma correcta no s'avisava de cap manera a l'usuari i es creava al directori l'arxiu contenidor de les pràctiques. Enllaç de descarrega de la primera versió: https://www.dropbox.com/s/15j657yu6eni67h/ETSEIB_Calendar_V0.apk?dl=0